RICE UNIVERSITY

Three-Dimensional, Time-Dependent Spectral Element Simulations of a Thermocapillary Liquid Bridge with Magnetic Stabilization

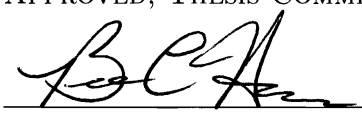
by

Kenneth E. Davis

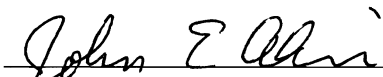
A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Master of Science

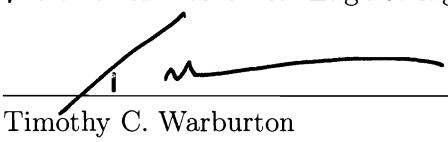
Approved, Thesis Committee:



Brent C. Houchens, Chair
Assistant Professor of Mechanical Engineering



John E. Akin
Professor of Mechanical Engineering



Timothy C. Warburton
Associate Professor of Computational and Applied Mathematics

Houston, Texas

April, 2011

ABSTRACT

Three-Dimensional, Time-Dependent Spectral Element Simulations of a Thermocapillary Liquid Bridge with Magnetic Stabilization

by

Kenneth E. Davis

The spectral element method is used to obtain 3D, time-dependent solutions for a thermocapillary driven liquid bridge with magnetic stabilization, which arises from the float-zone crystal growth process. The methods and implementation of the general, in-house developed fluid flow and heat transfer spectral element solver are discussed and the code is benchmarked. This work compares three-dimensional, time-dependent results to perturbations predicted by linear stability theory for the full-zone problem with Prandtl number of 0.02. Critical points, mode numbers, and azimuthal velocity perturbations are matched for the instabilities. Additionally, the simulations extend the study beyond the initial bifurcation point to find modal competition between two steady modes for the zero magnetic field case. Applying an axial magnetic field damps the perturbations and delays instabilities, providing a quiescent interior region that is conducive to growing defect free, uniform composition crystals. Weak magnetic fields are shown to remove the modal competition that leads to undesirable, time-dependent flow with mode switching.

Acknowledgments

I would first like to express my sincere gratitude to my advisor and thesis committee chair Dr. Brent Houchens for his guidance and support throughout my time at Rice.

I am grateful to Dr. Timothy Warburton and Dr. John E. Akin for serving on my thesis committee and offering advice for improving the quality of my thesis.

I have been fortunate to have Jacky Huang as an officemate. His willingness to always answer my questions and discuss ideas has been an invaluable part of my research process. I am also thankful to the other members of our group, Paul Boyle, Chad Kong, and Stephanie Tritchler, for all of their help.

I am grateful to my family for their constant encouragement of my studies.

Most of all, I want to thank my wife Batoul for always being supportive while I spent long hours working on my code and preparing my thesis.

This work was supported by the United States Air Force Office of Scientific Research and the National Science Foundation.

Contents

Abstract ii

Acknowledgments iii

List of Illustrations vi

List of Tables ix

1 Introduction and Motivation 1

1.1 Thermocapillary Driven Liquid Bridge 1

1.2 Half-Zone and Full-Zone Models 2

1.3 Magnetic Damping 3

1.4 Spectral Element Method 4

2 Thermocapillary Liquid Bridge Problem Formulation 6

2.1 Full-Zone Model 6

2.2 Weak Form of the Governing Equations 10

2.3 Spectral Element Discretization 12

2.4 Discrete System 15

2.4.1 Radial (r) Momentum 16

2.4.2 Azimuthal (θ) Momentum 19

2.4.3 Axial (z) Momentum 21

2.4.4 Continuity 22

2.4.5 Energy Equation 22

2.4.6 Conservation of Charge 23

3 Methods and Implementation 25

3.1 Meshing 25
3.2 Input File 29
3.3 Constructing the Spectral Mesh 33
3.4 Building Matrices 35
3.5 Solving the Matrix System 37
3.5.1 Solving the Linear System 39
3.5.2 Transient Solutions 42

4 Benchmarking **45**
4.1 Lid-Driven Cavity 46
4.2 Kovasznay Flow 47
4.3 Taylor Vortex 48
4.4 Half-Zone Model of the Liquid Bridge 51

5 Full-Zone Results **56**
5.1 The Critical Point 56
5.2 $Ha = 0$ Modes 59
5.3 Effects of Magnetic Stabilization 64

6 Conclusions and Future Work **70**
6.1 Conclusions 70
6.2 Future Work 71

Bibliography **73**

A Algorithms **77**
A.1 Preconditioned GMRES 77
A.2 Preconditioned BiCGStab 78
A.3 ILU(0) Preconditioner 79

Illustrations

2.1 Model of the full-zone with parabolic heat flux on the free surface and an axially applied magnetic field. 6

2.2 The nodal distribution on the local element for the 5th-order flow grid (solid) and the 3rd-order pressure grid (dashed). 15

3.1 Flow diagram for the solver. The dashed line encompasses the boxes that represent the code. Note that the geometry file is needed for the 2D version of the solver, but not the 3D solver. 26

3.2 A Gambit neutral file for the displayed mesh containing twelve nodes, two elements, and four boundaries. 27

3.3 Input file for the cylindrical mesher. 28

3.4 Input file for the solver. 30

3.5 A MATLAB spy plot (with color) of a Jacobian matrix built for the full-zone. The dashed lines form blocks that correspond to the entries in the Jacobian. 39

4.1 Lid driven cavity velocity profiles for $Re = 400$: (a) x (horizontal) velocity component along vertical centerline; (b) y (vertical) velocity component along a horizontal centerline. 46

4.2 Streamlines for the lid driven cavity flow projected onto the $z = 0.5$ plane. 47

4.3 Velocity vectors on the plane $\theta = \Theta$ for the cylindrical Kovasznay flow. 49

4.4 The taylor vortex velocity vectors. 50
4.5 Error of the Taylor vortex simulation at nondimensional time $t = 1.0$ against the order of time integration. 51
4.6 Plots on the $z = 0.5$ plane in the half-zone of a) the azimuthal velocity and b) the temperature. 53
4.7 Velocity vectors on a plane that bisects a) the hot spot and b) the cold spot. 54

5.1 The base flow streamlines on any $r - z$ plane in the cylinder. Note that the flow is symmetric about the $r = 0$ and $z = 0$ planes. 57
5.2 The azimuthal velocity contours for a) the anti-symmetric mode from linear stability theory, b) the anti-symmetric mode from 3D simulations, c) the symmetric mode from linear stability analysis, and d) the symmetric mode from 3D simulations. The contours from the 3D simulations for the anti-symmetric mode are at $1.91n$ for $n = -3, \ldots, 3$ and for the symmetric modes, the contours are at $1.33n$ for $n = -2, \ldots, 3$. 60
5.3 Time instances of the surface flow with $Ha = 0$ and $Re_{tc} = 1750$. The color represents the azimuthal velocity. The six instances cover the period of the of the cycle. 62
5.4 Maximum azimuthal velocity for the $Ha = 0$, $Re_{tc} = 1750$ case. . . . 63
5.5 The transition from the anti-symmetric $m = 2$ mode to the symmetric $m = 2$ mode occurs due to the break in symmetry across the axial midplane (represented by the black line). 64
5.6 Formation of cells in the base flow for a) $Ha = 0$, b) $Ha = 25$, and c) $Ha = 50$ all at $Re_{tc} = 1000$. The streamlines are colored by velocity magnitude on a logarithmic scale. 65

5.7 Maximum azimuthal velocity for $Ha = 5$ vs. a) nondimensional time with $Re_{tc} = 2100$ and b) thermocapillary Reynolds number. 66

5.8 The stationary $m = 2$ mode for $Ha = 5$ at $Re_{tc} = 1950$ shown in plots of a) the azimuthal velocity and b) electric potential. 67

5.9 A comparison of 3D simulations to linear stability theory for the $Ha = 5$, $m = 2$ modes via contours of a) the azimuthal velocity from linear stability, b) the azimuthal velocity from 3D simulations, c) the electric potential from linear stability, and d) the electric potential from 3D simulations. The azimuthal velocity from 3D simulations contours are at $1.13n$ for $n = -4, \ldots, 4$ and the electric potential contours are at 0.113 for $n = -7, \ldots, 6$. 68

5.10 The stationary $m = 2$ mode for $Ha = 10$ at $Re_{tc} = 2750$ shown in plots of a) the azimuthal velocity and b) electric potential. 68

5.11 A comparison of 3D simulations to linear stability theory for the $Ha = 10$, $m = 2$ modes via contours of a) the azimuthal velocity from linear stability, b) the azimuthal velocity from 3D simulations, c) the electric potential from linear stability, and d) the electric potential from 3D simulations. The contours from the 3D simulations for the azimuthal velocity are at 0.972 for $n = -6, \ldots, 6$ and the contours for electric potential are at 0.244 for $n = -3, \ldots, 3$. 69

Tables

3.1 Iterations, averaged over 7 Newton-Raphson steps, needed in solving the half-zone liquid bridge problem with $Re_{tc} = 1500$. Dashes represent divergence. 41

3.2 Total time elapsed while building preconditioners and solving the linear systems for the half-zone liquid bridge problem with $Re_{tc} = 1500$. Dashes represent divergence. 42

3.3 Weights for the Backwards Differentiation Formula. 43

4.1 Maximum values of the velocity magnitude and the azimuthal velocity component at $Re = 3500$ for different grids, which are defined by the number of nodes in each direction ($N_r \times N_\theta \times N_z$). . . 54

5.1 Comparison of $Re_{tc,cr}$ found from linear stability theory with the bounds found through 3D, time marching simulations. 58

Chapter 1

Introduction and Motivation

There are two goals to this thesis. The first is to obtain time-accurate solutions of a liquid bridge thermocapillary flow problem with magnetic stabilization, search for instabilities and compare to those predicted by linear stability theory. This is motivated by the melt region in optically heated floating-zone crystal growth processing. The second goal is to describe the general purpose fluid flow and heat transfer solver used to solve the thermocapillary flow problem.

1.1 Thermocapillary Driven Liquid Bridge

In the floating-zone crystal growth process, a polycrystalline rod is pulled through a heater, melted, and resolidified as a single crystal. The liquid bridge (the molten region) is held in place by surface tension, but gradients in surface tension due to temperature variations at the surface create a thermocapillary force that drives the flow in a toroidal manner. These toroidal cells are steady and axisymmetric if the temperature difference is sufficiently small. However, if the thermocapillary driving force becomes sufficiently large, the flow transitions to a steady or unsteady non-axisymmetric flow. Such circulatory flow may result in defects in the grown crystals. Periodic flows are particularly harmful as they result in periodic remelting at the

growth interface. To prevent such instabilities, the base state can be damped by applying a steady magnetic field parallel to the axis of the liquid bridge, inducing a current within the melt and resulting in Lorentz body forces that resist fluid movement in the radial and azimuthal directions. This leaves a mostly quiescent base state and, consequently, produces a crystal with uniform properties.

1.2 Half-Zone and Full-Zone Models

There are two prevailing models for studying the liquid bridge that forms in the float-zone process. They are the half-zone and the full-zone, named by the portion of the liquid bridge that is included in the model. The half-zone consists of a cylinder with an adiabatic free surface and rigid isothermal ends held at different temperatures, with the hotter end representing the axial mid-plane of the float-zone. The full-zone is also bounded by isothermal rigid disks, but both are at the resolidification temperature, and the free surface is subjected to radiative heating. The full-zone model attempts to capture the physics more accurately whereas the half-zone model assumes that the unphysical mid-plane boundary conditions (no-slip and isothermal) do not significantly affect the flow dynamics. Houchens and Walker investigated the two models using linear stability theory and found that the instability mechanisms are qualitatively similar, but are quantitatively different [1]. It is also worth noting that the assumed cylindrical shape for each model is not exact. When growing crystals in microgravity (as has been of experimental interest), the liquid bridge has a slight

bulge at the axial mid-plane. For this thesis, the cylindrical full-zone model is used.

1.3 Magnetic Damping

Studies on this problem assume low Prandtl numbers when modeling molten semiconductors, and use the nondimensional Hartmann number Ha and thermocapillary Reynolds number Re_{tc} to measure the strength of the magnetic field and the thermocapillary driving force, respectively. The goal is generally to locate the critical value of Re_{tc} for a given Ha. However, as Ha grows the flow becomes more stable, meaning much higher values of Re_{tc} are needed to trip the instability. With magnetic stabilization, multiple counter-rotating cells form, and the outermost rotational cell is characterized by high velocity gradients. Thus, the resolution necessary to adequately capture the flow dynamics increases quickly with increasing Ha. For $Pr = 0.02$, Houchens performed simulations up to $Ha = 100$ using a 4th-order stream function - temperature formulation to solve the axisymmetric the base flow [2]. However, increasing Ha further proved to be numerically challenging as the Chebyshev expansion of the 4th-order terms becomes very large, leaving the equations poorly balanced and susceptible to numerical error. Recently, Huang circumvented this problem by instead using a 2nd-order vorticity transport formulation to solve for the base flow and was able to reach $Ha = 300$ for $Pr = 0.02$ [3].

Here 3D time-dependent numerical flow simulations are used to find the critical Re_{tc} for a range of Ha, confirm the results obtained from linear stability theory,

and visualize the primary and secondary instabilities. These types of simulations have been performed for the half-zone without magnetic stabilization [4], but to the author's knowledge, there have been no extensive 3D simulations of the full-zone with magnetic stabilization.

1.4 Spectral Element Method

For accurate solutions of the full-zone problem, a spectral element method is implemented. The spectral element method was first used in 1984 by Patera [5] and is essentially a finite element method that uses high-order function approximations on elemental grids typical of spectral methods. With this approach, the advantages of the finite element method — approximations in complex geometries and basis functions with local support — are combined with the high-order accuracy and spectral convergence seen in spectral techniques.

For flow simulations, one must always be careful to satisfy the Babuska-Brezzi inf-sup condition to avoid spurious pressure modes [6, 7]. To satisfy this condition the stable $P_N - P_{N-2}$ Legendre elements have been adopted, on which the velocity is interpolated on the order N Gauss-Legendre-Lobatto nodes and the pressure on the order $N-2$ Gauss-Legendre nodes [8]. However, it is possible to use equal order interpolations for velocity and pressure by using properly formulated splitting schemes [9]. Here, the fully-coupled system of equations is solved.

In addition to solving for the flow field in the liquid bridge, this thesis details

the methods and techniques used within our in-house developed 3D flow and heat solver. The solver was written from scratch in Fortran 90 and originally intended to solve the base flow for the liquid bridge problem, but generality in the flow and thermal conditions was included during construction. However, the generality does not extend to Maxwell's equations; instead, the body forces that arise in the Navier-Stokes equations due to the axial magnetic field are hard-coded.

The main features of the code that lead to its generality are

- Choice of boundary conditions (e.g.: velocity, stress, temperature, heat flux)
- Buoyancy option with the Boussinesq approximation
- Spatial and temporal variability of boundary conditions
- Spatial variablity of initial condition

In this thesis, the governing equations for the full-zone model are derived, including the weak form of the equations, and these are discretized via the spectral element method in Chapter 2. The implementation details of the solver are discussed in Chapter 3. In Chapter 4, benchmarks are presented. New results for the liquid bridge problem are given in Chapter 5. Chapter 6 provides the conclusions and recommendations future work.

Chapter 2

Thermocapillary Liquid Bridge Problem Formulation

In this chapter the equations that govern the full-zone model are derived. The equations are first presented in their strong form, then in their weak form, and finally, in their discrete weak form.

2.1 Full-Zone Model

Figure 1 shows the full-zone model that was described in Chapter 1. The liquid

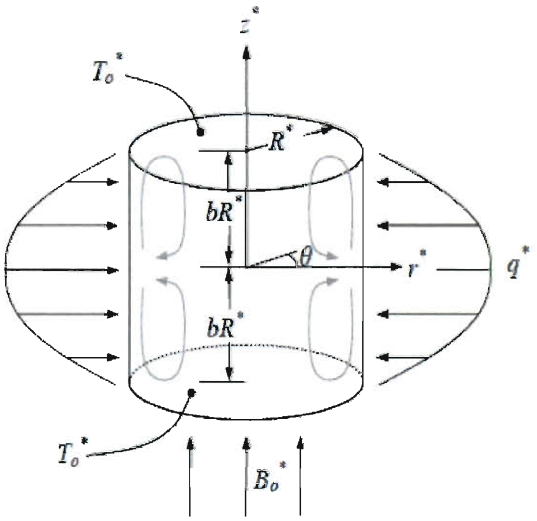


Figure 2.1 : Model of the full-zone with parabolic heat flux on the free surface and an axially applied magnetic field [10].

bridge is modeled as an undeformable cylinder with radius R^* and a height of $2bR^*$, where b is the aspect ratio. For the purposes of this thesis, $b = 1$ is set so it will not be present hereafter. Each end of the cylinder is isothermal and set to the resolidification temperature T_0^* while an external heater applies a parabolic, axisymmetric heat flux of the form

$$k^* \frac{\partial T^*}{\partial r^*} = q^* \left[1 - \left(\frac{z^*}{R^*} \right)^2 \right]$$

radially into the melt at the free surface, where k^* is the thermal conductivity, T^* is the temperature, q^* is the maximum heat flux value, and r^* and z^* are the dimensional cylindrical coordinates. There is an external constant magnetic field applied in the axial direction with magnetic flux density $B_0^* \hat{e}_z$. In this thesis, all variables with an asterisk are dimensional and those without asterisks are non-dimensional.

The appropriate scales for this problem are now selected. A viscous scale is chosen for the velocity such that $V^* = \mu^*/(\rho^* R^*)$, where μ^* is the absolute viscosity and ρ^* is the density of the melt. The scales for length, pressure, and time are R^*, $\rho^* \nu^{*2}/R^{*2}$, and R^{*2}/ν^*, respectively, where $\nu^* = \mu^*/\rho^*$ is the kinematic viscosity. The characteristic temperature scale is defined as $\Delta T^* = q^* R^*/k^*$ and the temperature is nondimensionalized using $(T^* - T_0^*)/\Delta T^*$, where T_0^* is the solidification temperature. The electric potential and electric current density scales are $V^* B_0^* R^*$ and $\sigma_e^* V^* B_0^*$, respectively, where σ_e^* is the electrical conductivity of the melt.

The nondimensional parameters Prandtl number and Hartmann number are de-

fined as

$$Pr = \frac{\mu^* c_p^*}{k^*} \quad \text{and} \quad Ha = \left(\frac{\sigma^*}{\mu^*}\right)^{\frac{1}{2}} B_0^* R^*,$$

where c_p^* is the specific heat of the melt. The magnetic Reynolds number, $Re_m = \mu_p^* \sigma_e^* V^* R^*$, where μ_p^* is the magnetic permeability of the molten region, is assumed to be small, which is true for semiconductor crystal growth applications. With this assumption the magnetic fields induced by electric currents in the melt can be neglected.

Now, the nondimensional Navier-Stokes, incompressible continuity, thermal energy, and conservation of charge equations can be written as

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = \nabla \cdot \sigma + Ha^2 \mathbf{f} \tag{2.1}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{2.2}$$

$$Pr \left[\frac{\partial T}{\partial t} + (\mathbf{u} \cdot \nabla)T\right] = \nabla^2 T \tag{2.3}$$

$$\nabla \cdot \mathbf{j} = 0 \tag{2.4}$$

where the stress tensor, electromagnetic body force, and Ohm's law for a purely vertical (axial) magnetic field are given, respectively, by

$$\sigma = -p\mathbf{I} + \nabla \mathbf{u} + (\nabla \mathbf{u})^T,$$

$$\mathbf{f} = \mathbf{j} \times \hat{\mathbf{z}},$$

and

$$\mathbf{j} = -\nabla \phi + \mathbf{u} \times \hat{\mathbf{z}},$$

where $\mathbf{j}$ is the electric current density and ϕ is the electric potential. At each end of the cylinder, the boundary conditions

$$\mathbf{u} = \mathbf{0}, \quad T = 0, \quad \text{and} \quad j_z = 0 \tag{2.5}$$

are applied. At the free surface the conditions

$$u_r = 0, \quad \frac{\partial T}{\partial z} = 1 - z^2, \quad \text{and} \quad j_r = 0, \tag{2.6}$$

are applied and two more conditions for the velocity or stress are needed. These conditions can be derived from the change in surface tension due to temperature variations and are the driving force in the flow. The surface tension is assumed to vary linearly with temperature such that

$$\gamma^* = \gamma_0^* + \frac{d\gamma^*}{dT^*}\left(T^* - T_0^*\right),$$

where $d\gamma^*/dT^*$ is a negative quantity. The thermocapillary conditions are then derived by balancing the surface tension gradient with the shear stress at the free surface. The axial and azimuthal directions yield the conditions

$$\tau_{rz}^* = \frac{d\gamma^*}{dz^*} = \frac{d\gamma^*}{dT^*}\frac{\partial T^*}{\partial z^*} \quad \text{and} \quad \tau_{\theta z}^* = \frac{1}{r^*}\frac{d\gamma^*}{d\theta^*} = \frac{1}{r^*}\frac{d\gamma^*}{dT^*}\frac{\partial T^*}{\partial \theta^*}.$$

The surface tension γ^* is scaled by $\left|\frac{d\gamma^*}{dT^*}\right| \Delta T^*$, which then allows for setting the value $\frac{d\gamma}{dT} = -1$. Completing the nondimensionalization gives the two final boundary conditions at the free surface

$$\tau_{rz} = -Re_{tc}\frac{dT}{dz} \quad \text{and} \quad \tau_{r\theta} = -Re_{tc}\frac{1}{r}\frac{dT}{d\theta} \tag{2.7}$$

where Re_{tc} is the thermocapillary Reynolds number defined by

$$Re_{tc} = \frac{\rho^* R^* \left|\frac{d\gamma^*}{dT^*}\right| \Delta T^*}{\mu^{*2}}.$$

This leaves equations (2.1) – (2.7) as the final form of the governing equations and boundary conditions.

2.2 Weak Form of the Governing Equations

The weak form of the governing equations are derived by first defining the following relevant spaces

$$\mathcal{X} = \left\{u \mid u \in H^1(\Omega),\ u|_\Gamma = u_D\right\}$$

$$\mathcal{V} = \left\{w \mid w \in H^1(\Omega),\ w|_\Gamma = 0\right\}$$

$$\mathcal{P} = \left\{q \mid q \in L^2(\Omega)\right\},$$

where Ω represents the domain over which the subspaces are defined, Γ is the boundary of Ω, and u_D is the specified value of u on Γ. Here $L^2(\Omega)$ is the space of all functions that are square integrable over Ω and $H^1(\Omega)$ is the space of functions in $L^2(\Omega)$ whose derivatives are also in $L^2(\Omega)$. Let $\mathcal{X}$ denote the space for all variables except pressure, $\mathcal{V}$ denote the space for weighting functions applied to each equation except continuity, and $\mathcal{P}$ be the space for both pressure and the remaining weighting function.

Dotting $\mathbf{w} \in \mathcal{V}^3$ with equation (2.1) gives

$$\int_\Omega \mathbf{w} \cdot \left[\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} - \nabla \cdot \sigma - Ha^2\mathbf{f})\right] d\Omega = \mathbf{0},$$

where $\mathbf{u} \in \mathcal{X}^3$ and $p \in \mathcal{P}$. This equation is manipulated by first integrating the stress divergence term by parts to account for any stress boundary conditions. This term becomes

$$-\int_\Omega \mathbf{w} \cdot (\nabla \cdot \sigma)\, d\Omega = \int_\Omega \nabla \mathbf{w} : \sigma \, d\Omega - \int_\Gamma \mathbf{w} \cdot (\mathbf{n} \cdot \sigma)\, d\Gamma$$

where the quantity $\mathbf{n} \cdot \sigma$ is given on boundary Γ. The boundary term will be addressed later. For now, the quantity $\sigma = -p\mathbf{I} + \left[\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right]$ is substituted to get

$$\begin{aligned} &\int_\Omega \nabla \mathbf{w} : -p\mathbf{I} + \nabla \mathbf{w} : \left[\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right] \, d\Omega \\ &= -\int_\Omega (\nabla \cdot \mathbf{w}) p \, d\Omega + \int_\Omega \nabla \mathbf{w} : \left[\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right] \, d\Omega. \end{aligned}$$

Combining yields the weak form of conservation of momentum

$$\begin{aligned} \int_\Omega \mathbf{w} \cdot \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} - Ha^2 \mathbf{f}\right) - (\nabla \cdot \mathbf{w}) p + \nabla \mathbf{w} : \left[\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right] \, d\Omega \\ = \int_\Gamma \mathbf{w} \cdot (\mathbf{n} \cdot \sigma)\, d\Gamma. \end{aligned} \tag{2.8}$$

Next, the continuity equation (2.2) is addressed. However, integration by parts is not necessary for this equation so the test function $q \in \mathcal{P}$ is simply multiplied by continuity, and integration is performed over the domain

$$\int_\Omega q(\nabla \cdot \mathbf{u})\, d\Omega = 0. \tag{2.9}$$

For the thermal energy equation (2.3), the conduction term must be integrated by parts in order to apply the heat flux condition. The test function $w_T \in \mathcal{V}$ is multiplied by the energy equation and integration is performed over the domain, to get

$$\int_\Omega w_T \left[Pr\left(\frac{\partial T}{\partial t} + (\mathbf{u} \cdot \nabla) T\right) - \nabla^2 T\right] = 0$$

where $T \in \mathcal{X}$. The integration by parts follows as

$$\int_\Omega -w_T \nabla^2 T \, d\Omega = \int_\Omega \nabla w_T \cdot \nabla T \, d\Omega - \int_\Gamma w_T (\nabla T \cdot \mathbf{n}) d\Gamma.$$

With this expression the weak form of the energy equation is

$$\int_\Omega Pr \, w_T \left(\frac{\partial T}{\partial t} + (\mathbf{u} \cdot \nabla) T \right) + \nabla w_T \cdot \nabla T \, d\Omega = \int_\Gamma w_T (\nabla T \cdot \mathbf{n}) d\Gamma. \tag{2.10}$$

Finally, the conservation of charge equation (2.4) is converted to its weak form. Multiplying by the weighting function $w_\phi \in \mathcal{V}$ and integrating over the domain yields

$$\int_\Omega w_\phi (\nabla \cdot \mathbf{j}) \, d\Omega = 0$$

where $\phi \in \mathcal{X}$. Integration by parts gives

$$\int_\Omega \nabla w_\phi \cdot \mathbf{j} \, d\Omega = \int_\Gamma w_\phi (\mathbf{j} \cdot \mathbf{n}) \, d\Gamma.$$

Now Ohm's law $\mathbf{j} = -\nabla \phi + \mathbf{u} \times \hat{\mathbf{z}}$ is substituted to get the weak form

$$\int_\Omega \nabla w_\phi \cdot (\mathbf{u} \times \hat{\mathbf{z}}) - \nabla w_\phi \cdot \nabla \phi \, d\Omega = \int_\Gamma w_\phi (\mathbf{j} \cdot \mathbf{n}) \, d\Gamma. \tag{2.11}$$

The complete statement of the weak form is now: given $\sigma \cdot \mathbf{n}$, $T \cdot \mathbf{n}$, and $\mathbf{j} \cdot \mathbf{n}$ on Γ, find u_r, u_θ, u_z, T, $\phi \in \mathcal{X}$ and $p \in \mathcal{P}$ satisfying (2.8)-(2.11) $\forall$ w_r, w_θ, w_z, w_T, $w_\phi \in \mathcal{V}$ and $\forall$ $q \in \mathcal{P}$.

2.3 Spectral Element Discretization

Next, the domain is discretized into K elements each denoted Ω_k such that

$$\bigcup_{k=1}^{K} \Omega_k = \Omega \quad \text{and} \quad \bigcap_{k=1}^{K} \Omega_k = \emptyset.$$

With this discretization, the discrete solutions now live in finite dimensional subspaces such that $\chi^\delta \in \mathcal{X}^\delta \subset \mathcal{X}$, $w_i^\delta \in \mathcal{V}^\delta \subset \mathcal{V}$, and $p^\delta, q^\delta \in \mathcal{P}^\delta \subset \mathcal{P}$ where χ and w_i represent variables and weighting functions in the spaces $\mathcal{X}$ and $\mathcal{V}$, respectively, and the superscript δ accounts for the approximations made through the creation of subdomains Ω_k and the interpolation over each element.

A general variable β is interpolated on each element using a linear expansion of basis functions ψ_i with compact support on the element,

$$\beta = \sum_{i}^{nn_e} c_i \psi_i(r, \theta, z), \tag{2.12}$$

where nn_e is the number of nodes on an element and c_i are constants. One convenient choice of basis functions is the set of Lagrange polynomials defined in one dimension as

$$\ell_{i,1D}(\xi) = \prod_{j=0, j\neq i}^{N} \frac{\xi - \xi_j}{\xi_i - \xi_j}$$

where N is the order of the polynomial and the subscripts i and j are used to specify particular nodes. The three-dimensional Lagrange polynomials are simply the tensor product of the one-dimensional polynomials. These polynomials have the nice property

$$\ell_i(r_j, \theta_j, z_j) = \delta_{ij},$$

which means that the ith Lagrange polynomial will have the value zero at all nodes other than the ith node, where it has the value one. This allows the expansion (2.12)

to take on the new form

$$\beta = \sum_{i}^{nn_e} \beta_i \ell_i(r, \theta, z) \tag{2.13}$$

where the coefficients are now the value of the function at that particular node. The expansion (2.13) is used for all variables in the derivation of the discrete system.

However, the set of nodes on which interpolation is performed have yet to be defined. In general, the choice of nodes depends on the type of element being used. For the purposes of this thesis, only hexahedral elements are considered because the optimal node set (those that minimize the Lebesgue constant) is known to be the tensor product of the one-dimensional Gauss-Legendre-Lobatto (GLL) nodes [11] , which are defined by $\xi_0 = -1$, $\xi_N = 1$, and the $(N-1)$ roots of $dP_N/d\xi$, where $P_N(\xi)$ is the Nth order Legendre polynomial. These nodes are used to interpolate velocity, temperature, electric potential, and the weighting functions w_r, w_θ, w_z, w_T, and w_ϕ. In this thesis these nodes are referred to as the "flow grid." However, as mentioned in Chapter 1, the Babuska-Brezzi condition must be satisfied to avoid spurious pressure modes. Thus, the pressure is interpolated with basis functions two orders lower than the velocity basis functions and on a different set of points known as the Gauss-Legendre (GL) nodes defined as the N roots of the Nth order Legendre polynomial. This is the "pressure grid." Note that these nodes lie on the interior of the element as can be seen in Figure 2.2. The weighting function for the continuity equation q is also interpolated over the pressure grid. When referring to the Lagrange polynomials defined by the pressure grid, ℓ_j^p is written. Because order N basis functions are used

for the flow grid and order $N-2$ for the pressure grid, this method is generally known as the P_N-P_{N-2} spectral element method.

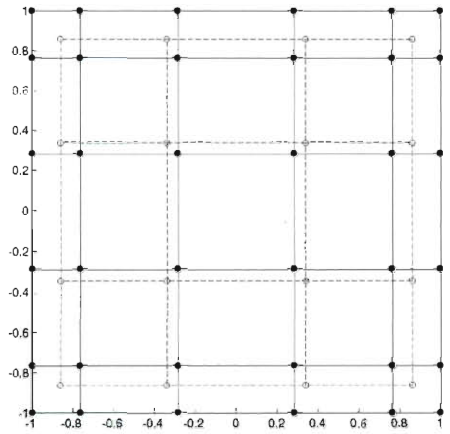


Figure 2.2 : The nodal distribution on the local element for the 5th-order flow grid (solid) and the 3rd-order pressure grid (dashed).

2.4 Discrete System

Using the discretization and interpolation of the previous section, the discrete system of equations can be derived.

Before proceeding to the derivation of the discrete equations, the required tensor calculus quantities are defined in cylindrical coordinates. The definitions for the gradient, divergence, and covariant derivative operators are

$$\nabla = \frac{\partial}{\partial r}\hat{\mathbf{r}} + \frac{1}{r}\frac{\partial}{\partial \theta}\hat{\boldsymbol{\theta}} + \frac{\partial}{\partial z}\hat{\mathbf{z}}$$

$$\nabla \cdot \mathbf{u} = \frac{1}{r}\frac{\partial r u_r}{\partial r} + \frac{1}{r}\frac{\partial u_\theta}{\partial \theta} + \frac{\partial u_z}{\partial z}$$

$$\nabla \mathbf{u} = \begin{bmatrix} \frac{\partial u_r}{\partial r} & \frac{1}{r}\frac{\partial u_r}{\partial \theta} - \frac{u_\theta}{r} & \frac{\partial u_r}{\partial z} \\ \frac{\partial u_\theta}{\partial r} & \frac{1}{r}\frac{\partial u_\theta}{\partial \theta} + \frac{u_r}{r} & \frac{\partial u_\theta}{\partial z} \\ \frac{\partial u_z}{\partial r} & \frac{1}{r}\frac{\partial u_z}{\partial \theta} & \frac{\partial u_z}{\partial z} \end{bmatrix}.$$

Also, the expanded form of the magnetic body force $\mathbf{f}$ is

$$\mathbf{f} = \mathbf{j} \times \hat{\mathbf{z}} = (-\nabla\phi + \mathbf{u} \times \hat{\mathbf{z}}) \times \hat{\mathbf{z}} = -\nabla\phi \times \hat{\mathbf{z}} - (\hat{\mathbf{z}} \cdot \hat{\mathbf{z}})\mathbf{u} + (\mathbf{u} \cdot \hat{\mathbf{z}})\hat{\mathbf{z}}$$

$$= -\nabla\phi \times \hat{\mathbf{z}} - \mathbf{u} + u_z\hat{\mathbf{z}}$$

where the triple vector product identity $(a \times b) \times c = -(b \cdot c)a + (a \cdot c)b$ has been utilized. This leads to the components of $\mathbf{f}$,

$$f_r = -\frac{1}{r}\frac{\partial \phi}{\partial \theta} - u_r, \;\; f_\theta = \frac{\partial \phi}{\partial r} - u_\theta, \;\; f_z = 0.$$

2.4.1 Radial (r) Momentum

Starting with (2.8), each component of linear momentum conservation is analyzed. Using the cylindrical vector calculus definitions, the r momentum equation in its weak discrete form is

$$\int_\Omega w_r^\delta \left[\frac{\partial u_r^\delta}{\partial t} + u_r^\delta \frac{\partial u_r^\delta}{\partial r} + u_\theta^\delta \left(\frac{1}{r}\frac{\partial u_r^\delta}{\partial \theta} - \frac{u_\theta^\delta}{r} \right) + u_z^\delta \frac{\partial u_r^\delta}{\partial z} + Ha^2 \frac{1}{r}\frac{\partial \phi^\delta}{\partial \theta} + Ha^2 u_r^\delta \right]$$
$$-\frac{1}{r}\frac{\partial (r w_r^\delta)}{\partial r} p + 2\frac{\partial w_r^\delta}{\partial r}\frac{\partial u_r^\delta}{\partial r} + \frac{1}{r}\frac{\partial w_r^\delta}{\partial \theta}\left(\frac{1}{r}\frac{\partial u_r^\delta}{\partial \theta} - \frac{u_\theta^\delta}{r} + \frac{\partial u_\theta^\delta}{\partial r} \right)$$
$$+\frac{\partial w_r^\delta}{\partial z}\left(\frac{\partial u_r^\delta}{\partial z} + \frac{\partial u_z^\delta}{\partial r} \right) d\Omega = 0.$$

Note that the superscript δ has been attached to each variable since the domain has been discretized. Also, the boundary integral has been removed because no stress

conditions exist for the r momentum in this problem. Now the derivative in the pressure term is expanded, the Lagrange expansions for w_r and all variables are substituted and like terms are collected to get

$$\sum_i \sum_j \left[\int_\Omega \ell_i \ell_j \, d\Omega\right] \dot{u}^\delta_{r,j} + \left[\int_\Omega \ell_i \frac{\partial \ell_j}{\partial r} \, d\Omega\right] u^\delta_{r,i} u^\delta_{r,j} + \left[\int_\Omega \frac{\ell_i}{r} \frac{\partial \ell_j}{\partial \theta} \, d\Omega\right] u^\delta_{\theta,i} u^\delta_{r,j}$$
$$- \left[\int_\Omega \frac{\ell_i \ell_j}{r} \, d\Omega\right] u^\delta_{\theta,i} u^\delta_{\theta,j} + \left[\int_\Omega \ell_i \frac{\partial \ell_j}{\partial z} \, d\Omega\right] u^\delta_{z,i} u^\delta_{r,j} + Ha^2 \left[\int_\Omega \frac{\ell_i}{r} \frac{\partial \ell_j}{\partial \theta} \, d\Omega\right] \phi^\delta_j$$
$$+ Ha^2 \left[\int_\Omega \ell_i \ell_j \, d\Omega\right] u^\delta_{r,j} - \left[\int_\Omega \left(\frac{\partial \ell_i}{\partial r} + \frac{\ell_i}{r}\right) \ell^p_j \, d\Omega\right] p^\delta_j$$
$$+ \left[\int_\Omega 2 \frac{\partial \ell_i}{\partial r} \frac{\partial \ell_j}{\partial r} + \frac{1}{r^2} \frac{\partial \ell_i}{\partial \theta} \frac{\partial \ell_j}{\partial \theta} + \frac{\partial \ell_i}{\partial z} \frac{\partial \ell_j}{\partial z} \, d\Omega\right] u^\delta_{r,j}$$
$$+ \left[\int_\Omega \frac{1}{r} \frac{\partial \ell_i}{\partial \theta} \left(\frac{\partial \ell_j}{\partial r} - \frac{\ell_j}{r}\right) \, d\Omega\right] u^\delta_{\theta,j} + \left[\int_\Omega \frac{\partial \ell_i}{\partial z} \frac{\partial \ell_j}{\partial r} \, d\Omega\right] u^\delta_{z,j} = 0.$$

The summation indices are used to build matrices from the bracketed terms and vectors from the variables outside the brackets. This yields the matrix equation

$$\mathbf{M_r} \dot{\tilde{U}}_r + \tilde{U}_r \cdot \mathbf{C_{rr}} \tilde{U}_r + \tilde{U}_\theta \cdot \mathbf{C_{r\theta}} \tilde{U}_r - \tilde{U}_\theta \cdot \mathbf{C^\theta_{r\theta}} \tilde{U}_\theta + \tilde{U}_z \cdot \mathbf{C_{rz}} \tilde{U}_r$$

$$+ Ha^2 \mathbf{C_{r\phi}} \Phi + Ha^2 \mathbf{M_r} \tilde{U}_r - \mathbf{G_r} P + \mathbf{K_{rr}} \tilde{U}_r + \mathbf{K_{r\theta}} \tilde{U}_\theta + \mathbf{K_{rz}} \tilde{U}_z = 0.$$

The vectors with tildes include the values at every node in the domain; however, since Dirichlet boundary conditions are applied on a portion or all of the boundary, the values at the nodes corresponding to these conditions are known and, hence, do not need to be solved. Thus, we can separate the unknowns from the Dirichlet conditions by writing, for instance,

$$\tilde{U}_r = U_r + U_{r,D}$$

where U_r represents the vector of unknown radial velocities and $U_{r,D}$ represents a vector containing values for u_r on given boundaries. Substituting this sum into the matrix equation gives

$$\mathbf{M_r}\dot{U}_r + \tilde{U}_r \cdot \mathbf{C_{rr}}\tilde{U}_r + \tilde{U}_\theta \cdot \mathbf{C_{r\theta}}\tilde{U}_r - \tilde{U}_\theta \cdot \mathbf{C}^{\theta}_{\mathbf{r\theta}}\tilde{U}_\theta + \tilde{U}_z \cdot \mathbf{C_{rz}}\tilde{U}_r$$

$$+Ha^2\mathbf{D_{r\phi}}\Phi + Ha^2\mathbf{M_r}U_r - \mathbf{G_r}P + \mathbf{K_{rr}}U_r + \mathbf{K_{r\theta}}U_\theta + \mathbf{K_{rz}}U_z = F_r \tag{2.14}$$

where

$$F_r = -\mathbf{M_r}\dot{U}_r - Ha^2\mathbf{M_r}U_{r,D} - \mathbf{K_{rr}}U_{r,D} - \mathbf{K_{r\theta}}U_{\theta,D} - \mathbf{K_{rz}}U_{z,D} = 0.$$

Equation (2.14) is the final matrix form of the r momentum, but there are three items to notice here. First, the convective terms still contain the Dirichlet boundary contributions because the nonlinearity prevents the simple separation seen in the linear case. However, for this problem all Dirichlet conditions are homogeneous so that $\tilde{U_m} = U_m$ where m is either r, θ, or z, and the resulting equation is much simpler. Going forward, the tilde variables are kept as if the Dirichlet conditions were inhomogeneous in order to provide a more general framework. Secondly, there is no occurance of $\tilde{P}$ or $\tilde{\Phi}$ because there are no conditions explicitly applied to pressure or electric potential; instead, conditions are applied implicitly on the two variables through the stress conditions and the electric current density conditions, respectively. Lastly, all terms in F_r drop out because all of the Dirichlet conditions for this problem are homogeneous.

2.4.2 Azimuthal (θ) Momentum

The momentum in the θ direction is analyzed next. Starting with equation (2.8) and using the cylindrical vector calculus definitions gives

$$\int_\Omega w_\theta^\delta \left[\frac{\partial u_\theta^\delta}{\partial t} + u_r^\delta \frac{\partial u_\theta^\delta}{\partial r} + u_\theta^\delta \left(\frac{1}{r}\frac{\partial u_\theta^\delta}{\partial r} + \frac{u_r^\delta}{r}\right) + u_z^\delta \frac{\partial u_\theta^\delta}{\partial z} - Ha^2 \frac{\partial \phi^\delta}{\partial r} + Ha^2 u_\theta^\delta\right]$$
$$-\frac{1}{r}\frac{\partial w_\theta^\delta}{\partial \theta} p^\delta + \frac{\partial w_\theta^\delta}{\partial r}\left(\frac{\partial u_\theta^\delta}{\partial r} + \frac{1}{r}\frac{\partial u_r^\delta}{\partial \theta} - \frac{u_\theta^\delta}{r}\right) + \frac{2}{r}\frac{\partial w_\theta^\delta}{\partial \theta}\left(\frac{1}{r}\frac{\partial u_\theta^\delta}{\partial \theta} + \frac{u_r^\delta}{r}\right)$$
$$+\frac{\partial w_\theta^\delta}{\partial z}\left(\frac{\partial u_\theta^\delta}{\partial z} + \frac{1}{r}\frac{\partial u_z^\delta}{\partial \theta}\right) d\Omega = \int_\Gamma w_\theta^\delta (\sigma^\delta \cdot \mathbf{n})_\theta \, d\Gamma.$$

The approach here is the same as it was for the r momentum with one significant addition: there is now a stress condition to apply. In most applications the stress would be constant or possibly dependent on space or time, all of which would maintain the position of the term on the right-hand side of the equation. Here the stress depends on a temperature derivative and since temperature is an unknown, the boundary condition must be on the left-hand side. Considering only the Γ integral, where in this case Γ is the free surface, gives

$$\int_\Gamma w_\theta^\delta (\sigma^\delta \cdot \mathbf{n})_\theta \, d\Gamma = \int_\Gamma w_\theta^\delta \left(-Re_{tc}\frac{1}{r}\frac{\partial T^\delta}{\partial \theta}\right) d\Gamma = -Re_{tc}\left[\int_\Gamma \frac{\ell_i}{r}\frac{\partial \ell_j}{\partial \theta}\, d\Gamma\right] T_j^\delta.$$

Inserting this result, substituting in the approximations for the weighting function and each variable, and collecting like terms yields

$$\left[\int_\Omega \ell_i\ell_j\,d\Omega\right]\dot{u}^\delta_{\theta,j}+\left[\int_\Omega \ell_i\frac{\partial\ell_j}{\partial r}\,d\Omega\right]u^\delta_{r,i}u^\delta_{\theta,j}+\left[\int_\Omega \frac{\ell_i}{r}\frac{\partial\ell_j}{\partial\theta}\,d\Omega\right]u^\delta_{\theta,i}u^\delta_{\theta,j}+\left[\int_\Omega \frac{\ell_i\ell_j}{r}\,d\Omega\right]u^\delta_{\theta,i}u^\delta_{r,j}$$
$$+\left[\int_\Omega \ell_i\frac{\partial\ell_j}{\partial z}\,d\Omega\right]u^\delta_{z,i}u^\delta_{\theta,j}-Ha^2\left[\int_\Omega \ell_i\frac{\partial\ell_j}{\partial r}\,d\Omega\right]\phi^\delta_j+Ha^2\left[\int_\Omega \ell_i\ell_j\,d\Omega\right]u^\delta_{\theta,j}$$
$$-\left[\int_\Omega \frac{\partial\ell_i}{\partial\theta}\frac{\ell^p_j}{r}\,d\Omega\right]p^\delta_j+\left[\int_\Omega\left(\frac{1}{r}\frac{\partial\ell_i}{\partial r}\frac{\partial\ell_j}{\partial\theta}+2\frac{\partial\ell_i}{\partial\theta}\frac{\ell_j}{r^2}\right)d\Omega\right]u^\delta_{r,j}$$
$$+\left[\int_\Omega\left(\frac{\partial\ell_i}{\partial r}\frac{\partial\ell_j}{\partial r}-\frac{\partial\ell_i}{\partial r}\frac{\ell_j}{r}+\frac{2}{r^2}\frac{\partial\ell_i}{\partial\theta}\frac{\partial\ell_j}{\partial\theta}+\frac{\partial\ell_i}{\partial z}\frac{\partial\ell_j}{\partial z}\right)d\Omega\right]u^\delta_{\theta,j}$$
$$+\left[\int_\Omega \frac{1}{r}\frac{\partial\ell_i}{\partial z}\frac{\partial\ell_j}{\partial\theta}\,d\Omega\right]u^\delta_{z,j}+Re_{tc}\left[\int_\Gamma \frac{\ell_i}{r}\frac{\partial\ell_j}{\partial\theta}\,d\Gamma\right]T^\delta_j=0.$$

Converting to a matrix equation gives

$$\mathbf{M}_\theta\dot{\tilde{U}}_\theta+\tilde{U}_r\cdot\mathbf{C}_{\theta\mathbf{r}}\tilde{U}_\theta+\tilde{U}_\theta\cdot\mathbf{C}_{\theta\theta}\tilde{U}_\theta+\tilde{U}_\theta\cdot\mathbf{C}^{\mathbf{r}}_{\theta\theta}\tilde{U}_r+\tilde{U}_z\cdot\mathbf{C}_{\theta\mathbf{z}}\tilde{U}_\theta$$

$$-Ha^2\mathbf{D}_{\theta\phi}\Phi+Ha^2\mathbf{M}_\theta\tilde{U}_\theta-\mathbf{G}_\theta P+\mathbf{K}_{\theta\mathbf{r}}\tilde{U}_r+\mathbf{K}_{\theta\theta}\tilde{U}_\theta+\mathbf{K}_{\theta\mathbf{z}}\tilde{U}_z+Re_{tc}\mathbf{B}_\theta\tilde{T}=0.$$

Once again vectors with tildes are separated into known and unknown vectors to get the final θ momentum equations

$$\mathbf{M}_\theta\dot{U}_\theta+\tilde{U}_r\cdot\mathbf{C}_{\theta\mathbf{r}}\tilde{U}_\theta+\tilde{U}_\theta\cdot\mathbf{C}_{\theta\theta}\tilde{U}_\theta+\tilde{U}_\theta\cdot\mathbf{C}^{\mathbf{r}}_{\theta\theta}\tilde{U}_r+\tilde{U}_z\cdot\mathbf{C}_{\theta\mathbf{z}}\tilde{U}_\theta$$

$$-Ha^2\mathbf{C}_{\theta\phi}\Phi+Ha^2\mathbf{M}_\theta U_\theta-\mathbf{G}_\theta P+\mathbf{K}^T_{\mathbf{r}\theta}U_r+\mathbf{K}_{\theta\theta}U_\theta+\mathbf{K}_{\theta\mathbf{z}}U_z+Re_{tc}\mathbf{B}_\theta T=F_\theta. \quad (2.15)$$

where

$$F_\theta=-\mathbf{M}_\theta\dot{U}_{\theta,D}-Ha^2\mathbf{M}_\theta U_{\theta,D}-\mathbf{K}^T_{\mathbf{r}\theta}U_{r,D}-\mathbf{K}_{\theta\theta}U_{\theta,D}-\mathbf{K}_{\theta\mathbf{z}}U_{z,D}-Re_{tc}\mathbf{B}_\theta T_D=0.$$

Again, variables with tildes remain in the convective terms and all terms in F_θ drop out. Note that the simplification $\mathbf{K}_{\theta\mathbf{r}}=\mathbf{K}^T_{\mathbf{r}\theta}$ has been substituted into the final form.

2.4.3 Axial (z) Momentum

The derivation of the matrix equations for the momentum in the z direction is quite similar to the previous two momentum equations. The weak form is

$$\int_\Omega w_z^\delta \left[\frac{\partial u_z^\delta}{\partial t} + u_r^\delta \frac{\partial u_z^\delta}{\partial r} + u_\theta^\delta \frac{1}{r}\frac{\partial u_z^\delta}{\partial r} + u_z^\delta \frac{\partial u_z^\delta}{\partial z}\right]$$
$$-\frac{\partial w_z^\delta}{\partial z} p^\delta + \frac{\partial w_z^\delta}{\partial r}\left(\frac{\partial u_z^\delta}{\partial r} + \frac{\partial u_r^\delta}{\partial z}\right) + \frac{1}{r}\frac{\partial w_z^\delta}{\partial \theta}\left(\frac{1}{r}\frac{\partial u_z^\delta}{\partial \theta} + \frac{\partial u_\theta^\delta}{\partial z}\right)$$
$$+2\frac{\partial w_z^\delta}{\partial z}\frac{\partial u_z^\delta}{\partial z} d\,\Omega = \int_\Gamma w_z^\delta (\sigma^\delta \cdot \mathbf{n})_z \, d\Gamma.$$

The appropriate substitutions lead to

$$\sum_i \sum_j \left[\int_\Omega \ell_i \ell_j \, d\Omega\right] \dot{u}_{z,j}^\delta + \left[\int_\Omega \ell_i \frac{\partial \ell_j}{\partial r} \, d\Omega\right] u_{r,i}^\delta u_{z,j}^\delta + \left[\int_\Omega \frac{\ell_i}{r}\frac{\partial \ell_j}{\partial \theta} \, d\Omega\right] u_{\theta,i}^\delta u_{z,j}^\delta$$
$$+ \left[\int_\Omega \ell_i \frac{\partial \ell_j}{\partial z} \, d\Omega\right] u_{z,i}^\delta u_{z,j}^\delta - \left[\int_\Omega \frac{\partial \ell_i}{\partial z} \ell_j^p \, d\Omega\right] p_j^\delta + \left[\int_\Omega \frac{\partial \ell_i}{\partial r}\frac{\partial \ell_j}{\partial z} \, d\Omega\right] u_{r,j}^\delta + \left[\int_\Omega \frac{1}{r}\frac{\partial \ell_i}{\partial \theta}\frac{\partial \ell_j}{\partial z} \, d\Omega\right] u_{\theta,j}^\delta$$
$$+ \left[\int_\Omega \left(\frac{\partial \ell_i}{\partial r}\frac{\partial \ell_j}{\partial r} + \frac{1}{r^2}\frac{\partial \ell_i}{\partial \theta}\frac{\partial \ell_j}{\partial \theta} + 2\frac{\partial \ell_i}{\partial z}\frac{\partial \ell_j}{\partial z}\right) d\Omega\right] u_{z,j}^\delta + Re_{tc}\left[\int_\Gamma \ell_i \frac{\partial \ell_j}{\partial z} \, d\Gamma\right] T_j^\delta = 0.$$

In matrix form with the Dirichlet conditions separated from the unknowns, this becomes

$$\mathbf{M_z}\dot{U}_z + \tilde{U}_r \cdot \mathbf{C_{zr}}\tilde{U}_z + \tilde{U}_\theta \cdot \mathbf{C_{z\theta}}\tilde{U}_z + \tilde{U}_z \cdot \mathbf{C_{zz}}\tilde{U}_z$$
$$-\mathbf{G_z}P + \mathbf{K_{rz}^T}U_r + \mathbf{K_{\theta z}^T}U_\theta + \mathbf{K_{zz}}U_z + Re_{tc}\mathbf{B_z}T = F_z. \tag{2.16}$$

where

$$F_z = -\mathbf{M_z}\dot{U}_{z,D} - \mathbf{K_{rz}^T}U_{r,D} - \mathbf{K_{\theta z}^T}U_{\theta,D} - \mathbf{K_{zz}}U_{z,D} - Re_{tc}\mathbf{B_z}T_D = 0.$$

This completes the derivation of the matrix equations for the momentum equations.

2.4.4 Continuity

The continuity equation is analyzed much like the momentum equations. The weak form is

$$\int_\Omega q^\delta \left(\frac{1}{r}\frac{\partial (r u_r^\delta)}{\partial r} + \frac{1}{r}\frac{\partial u_\theta^\delta}{\partial \theta} + \frac{\partial u_z^\delta}{\partial z} \right) d\Omega = \int_\Omega q^\delta \left(\frac{\partial u_r^\delta}{\partial r} + \frac{u_r^\delta}{r} + \frac{1}{r}\frac{\partial u_\theta^\delta}{\partial \theta} + \frac{\partial u_z^\delta}{\partial z} \right) d\Omega = 0.$$

Substitution yields

$$\sum_i \sum_j \left[\int_\Omega \ell_i^p \left(\frac{\partial \ell_j}{\partial r} + \frac{\ell_j}{r} \right) d\Omega \right] u_{r,j}^\delta + \left[\int_\Omega \frac{\ell_i^p}{r}\frac{\partial \ell_j}{\partial \theta}\, d\Omega \right] u_{\theta,j}^\delta + \left[\int_\Omega \ell_i^p \frac{\partial \ell_j}{\partial z}\, d\Omega \right] u_{z,j}^\delta = 0$$

and building matrices gives

$$\mathbf{G_r}^T \tilde{U}_r + \mathbf{G}_\theta{}^T \tilde{U}_\theta + \mathbf{G_z}^T \tilde{U}_z = 0.$$

The final matrix equation for continuity is

$$\mathbf{G_r}^T U_r + \mathbf{G}_\theta^T U_\theta + \mathbf{G_z}^T U_z = F_p \tag{2.17}$$

where

$$F_p = -\mathbf{G_r}^T U_{r,D} - \mathbf{G}_\theta^T U_{\theta,D} - \mathbf{G_z}^T U_{z,D} = 0.$$

The absence of nonlinear terms means there are no Dirichlet boundary terms on the left hand side. Also, notice that the matrices are all just the transposes of matrices attached to the pressure term in the momentum equations.

2.4.5 Energy Equation

A similar analysis gives

$$\int_\Omega Pr\, w_T^\delta \left(\frac{\partial T^\delta}{\partial t} + u_r^\delta \frac{\partial T^\delta}{\partial r} + \frac{u_\theta^\delta}{r}\frac{\partial T^\delta}{\partial \theta} + u_z^\delta \frac{\partial T^\delta}{\partial z} \right) + \frac{\partial w_T^\delta}{\partial r}\frac{\partial T^\delta}{\partial r}$$
$$+ \frac{1}{r^2}\frac{\partial w_T^\delta}{\partial \theta}\frac{\partial T^\delta}{\partial \theta} + \frac{\partial w_T^\delta}{\partial z}\frac{\partial T^\delta}{\partial z}\, d\Omega = \int_\Gamma w_T^\delta (\nabla T^\delta \cdot \mathbf{n})\, d\Gamma.$$

With substitutions, we have

$$\sum_i \sum_j Pr \Bigg\{ \left[\int_\Omega \ell_i \ell_j \, d\Omega \right] \dot{T}_j^\delta + \left[\int_\Omega \ell_i \frac{\partial \ell_j}{\partial r}\, d\Omega \right] u_{r,i}^\delta T_j^\delta + \left[\int_\Omega \frac{\ell_i}{r}\frac{\partial \ell_j}{\partial \theta}\, d\Omega \right] u_{\theta,i}^\delta T_j^\delta$$
$$+ \left[\int_\Omega \ell_i \frac{\partial \ell_j}{\partial z}\, d\Omega \right] u_{z,i}^\delta T_j^\delta \Bigg\} + \left[\int_\Omega \frac{\partial \ell_i}{\partial r}\frac{\partial \ell_j}{\partial r} + \frac{1}{r^2}\frac{\partial \ell_i}{\partial \theta}\frac{\partial \ell_j}{\partial \theta} + \frac{\partial \ell_i}{\partial z}\frac{\partial \ell_j}{\partial z}\, d\Omega \right] T_j^\delta$$
$$= \sum_i \int_\Gamma \ell_i (\nabla T^\delta \cdot \mathbf{n})\, d\Gamma.$$

The final matrix equation is

$$Pr \left(\mathbf{M_T} \dot{T} + \tilde{U}_r \cdot \mathbf{C_{Tr}} \tilde{T} + \tilde{U}_\theta \cdot \mathbf{C_{T\theta}} \tilde{T} + \tilde{U}_z \cdot \mathbf{C_{Tz}} \tilde{T} \right) + \mathbf{K_T} T = F_T \tag{2.18}$$

where

$$F_T = \int_\Gamma \ell (\nabla T^\delta \cdot \mathbf{n})\, d\Gamma - Pr \mathbf{M_T} \dot{T}_D - \mathbf{K_T} T_D = \int_\Gamma \ell \left(1 - z^2\right)\, d\Gamma.$$

In this case the boundary Γ is the free surface and the applied heat flux is spatially varying.

2.4.6 Conservation of Charge

Finally, the weak form of the conservation of charge is

$$\int_\Omega \frac{\partial w_\phi^\delta}{\partial r} \left(u_\theta^\delta - \frac{\partial \phi^\delta}{\partial r} \right) - \frac{1}{r}\frac{\partial w_\phi^\delta}{\partial \theta} \left(u_r^\delta + \frac{1}{r}\frac{\partial \phi^\delta}{\partial \theta} \right) - \frac{\partial w_\phi^\delta}{\partial z}\frac{\partial \phi^\delta}{\partial z}\, d\Omega = \int_\Gamma w_\phi^\delta (\mathbf{j}^\delta \cdot \mathbf{n})\, d\Gamma$$

and by substituting in the appropriate expansions, we get

$$\sum_i \sum_j \left[\int_\Omega \frac{\partial \ell_i}{\partial r} \ell_j \, d\Omega \right] u^\delta_{\theta,j} - \left[\int_\Omega \frac{\partial \ell_i}{\partial \theta} \frac{\ell_j}{r} \, d\Omega \right] u^\delta_{r,j}$$

$$- \left[\int_\Omega \frac{\partial \ell_i}{\partial r} \frac{\partial \ell_j}{\partial r} + \frac{1}{r^2} \frac{\partial \ell_i}{\partial \theta} \frac{\partial \ell_j}{\partial \theta} + \frac{\partial \ell_i}{\partial z} \frac{\partial \ell_j}{\partial z} \, d\Omega \right] \phi^\delta_j = \sum_i \int_\Gamma \ell_i (\mathbf{j}^\delta \cdot \mathbf{n}) \, d\Gamma.$$

The final equation is

$$\mathbf{D}^T_{\theta\phi} U_\theta - \mathbf{D}^T_{\mathbf{r}\phi} U_r - \mathbf{K}_\phi \Phi = F_\phi \tag{2.19}$$

where

$$F_\phi = \int_\Gamma \ell (\mathbf{j} \cdot \mathbf{n}) \, d\Gamma - \mathbf{D}^T_{\theta\phi} U_{\theta,D} + \mathbf{D}^T_{\mathbf{r}\phi} U_{r,D} = 0.$$

In addition to the homogeneous Dirichlet boundary conditions, F_p contains the weak implementation of the current density condition. For this problem the current density normal to the boundary is zero so this term drops out as well.

The complete matrix system resulting from the spectral element discretization is (2.14) – (2.19).

Chapter 3

Methods and Implementation

The matrix system of the previous chapter must be built, assembled, and solved for the nodal values of each variable. This chapter provides a description of the solver, its usage, and the methods that are implemented within it. The source code is written in Fortran 90 and compiled with Intel's *ifort*. It makes use of subroutines provided in Intel's Math Kernel Library (MKL). A flow diagram for the solver is given in Figure 3.1.

3.1 Meshing

To use the solver, two inputs are needed: a mesh file and an input file containing all necessary parameter specifications for a given simulation. The former is discussed first. The initial step in the simulation process is to build a geometry and apply sufficiently fine discretization on which the solution can be approximated. The current version of the code supports hexahedral elements only, so it is necessary to use mesh software that can satisfy this limitation. Presently, all Cartesian geometries and meshes are created using Gambit 2.4, which is accessed through the TeraGrid [12]. With Gambit, the user can create a geometry, mesh using hexahedral elements, and label each face or a group of faces with a boundary number. The mesh is then exported

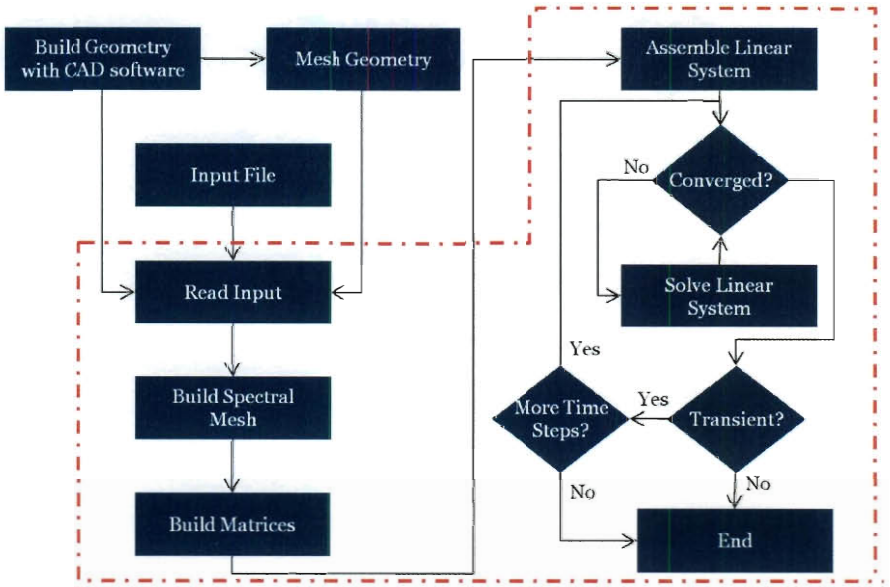


Figure 3.1 : Flow diagram for the solver. The dashed line encompasses the boxes that represent the code. Note that the geometry file is needed for the 2D version of the solver, but not the 3D solver.

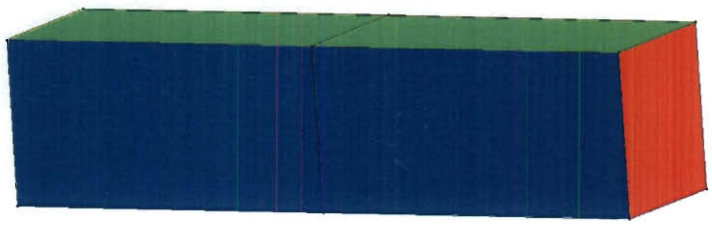
as a Gambit neutral file, which is read in by the code. An example of a neutral file and its corresponding mesh are given in Figure 3.1. The solver extracts three important sets of data from the neutral file: the three coordinates for each node, the eight nodes that make up the connectivity for each element, and the element number and face number that correspond to each boundary face. The mesh in Figure 3.2(b) shows the two elements and three of the boundary conditions, where red denotes the first boundary, green the second, and blue the third. The remaining face that is hidden from view accounts for the fourth boundary.

For problems more suitably formulated and solved in cylindrical coordinates, an in-house structured cylindrical mesher has been developed. A structured mesh is used

```
        CONTROL INFO 2.4.6
** GAMBIT NEUTRAL FILE
default_id2440
PROGRAM:                Gambit     VERSION:  2.4.6

     NUMNP     NELEM     NGRPS    NBSETS     NDFCD     NDFVL
        12         2         1         4         3         3
ENDOFSECTION
   NODAL COORDINATES 2.4.6
         1  1.00000000000e+000  0.00000000000e+000  4.00000000000e+000
         2  1.00000000000e+000  1.00000000000e+000  4.00000000000e+000
         3  0.00000000000e+000  1.00000000000e+000  4.00000000000e+000
         4  0.00000000000e+000  0.00000000000e+000  4.00000000000e+000
         5  1.00000000000e+000  1.00000000000e+000  0.00000000000e+000
         6  1.00000000000e+000  0.00000000000e+000  0.00000000000e+000
         7  0.00000000000e+000  0.00000000000e+000  0.00000000000e+000
         8  0.00000000000e+000  1.00000000000e+000  0.00000000000e+000
         9  0.00000000000e+000  0.00000000000e+000  2.00000000000e+000
        10  1.00000000000e+000  0.00000000000e+000  2.00000000000e+000
        11  0.00000000000e+000  1.00000000000e+000  2.00000000000e+000
        12  1.00000000000e+000  1.00000000000e+000  2.00000000000e+000
ENDOFSECTION
      ELEMENTS/CELLS 2.4.6
       1  4  8        1       2       10       12        4        3        9
                     11
       2  4  8       10      12        6        5        9       11        7
                      8
ENDOFSECTION
      ELEMENT GROUP 2.4.6
GROUP:       1 ELEMENTS:       2 MATERIAL:       2 NFLAGS:       1
                               fluid
       0
       1       2
ENDOFSECTION
 BOUNDARY CONDITIONS 2.4.6
                                 1        1        1        0        6
         2    4    3
ENDOFSECTION
 BOUNDARY CONDITIONS 2.4.6
                                 2        1        4        0        6
         1    4    2
         2    4    2
         1    4    4
         2    4    4
ENDOFSECTION
 BOUNDARY CONDITIONS 2.4.6
                                 3        1        4        0        6
         1    4    5
         2    4    5
         1    4    6
         2    4    6
ENDOFSECTION
 BOUNDARY CONDITIONS 2.4.6
                                 4        1        1        0        6
         1    4    1
ENDOFSECTION
```

(a) A neutral file.



(b) The physical mesh.

Figure 3.2 : A Gambit neutral file for the displayed mesh containing twelve nodes, two elements, and four boundaries.

```
Cylinder Bounds
-----------------
R          0.0 1.0
Th/Pi      0.0 2.0
Z          -1.0 1.0

Mesh Options
------------------
RThZels    10 10 20
Mesh_File cyl.neu

Boundary Layer Options
-------------------------
R Layer
-------------------------
end        outer
num_els    4
geo_ratio 1.2
first_el   0.05

Z Layer
-------------------------
end        both
num_els    4
geo_ratio 1.2
first_el   0.05

Boundary Numbering
-------------------------
Bottom Top Outer Inner Theta1 Theta2
-------------------------------------
1 2 3 4 5 6
```

Figure 3.3 : Input file for the cylindrical mesher.

because of the difficulty in integrating around the singularity at $r = 0$. This mesher forces nodes to be placed at $r = 0$ and integration via Gauss quadrature is computed using interior points (see Section 3.4).

To use the cylindrical mesher, the input file in Figure 3.3 must be present. As the file suggests, the first section defines the minimum and maximum values for the variables where $0 \leq r < \infty$, $0 \leq \theta/\pi \leq 2$, and $-\infty < z < \infty$. The second section specifies the number of elements in each direction and the name of the file to be created. Next are the boundary layer options. For the radial direction, a layer can be placed on the *inner* or *outer* boundary, or *both* can be specified. Similarly, the z boundary layer *end* options are *bottom*, *top*, or *both*. For each layer, the number of elements in the layer N_{bl}, the geometric ratio r_g, and the first element size d_0 must

be selected. With these parameters, the element sizes are defined as

$$d_i = r_g d_{i-1} = r_g^i d_0, \quad \text{for } i = 1, 2, ..., N_{bl} - 1.$$

The final section of the input file specifies the order in which the boundaries are numbered. This is important because the code gives the boundary with the lowest number the highest priority. For instance, in the case of the lid driven cavity flow, there is a discontinuity where the lid edge meets the container wall in the typical definition of the problem. Since the velocity solution is in a continuous space, the discontinuity is smoothed. The user must choose where this regularization will take place (either on the lid or on the container walls) by specifying either a no-slip condition or the lid velocity for this edge.

Running the cylindrical mesher produces a neutral file similar to those exported from Gambit.

3.2 Input File

Besides the mesh file, the only other input required by the current version of the code is an input file. This file specifies the problem and solution parameters. A typical input file is shown in Figure 3.4. The problem being set up in this figure is the lid driven cavity problem with a velocity at the lid that is linear in time until $t = 1$, after which the velocity is constant.

In the first section the user specifies the physics that are involved by setting the *problem type*, the coordinate frame, and whether the simulation is steady or transient.

```
-----------RUN PARAMETERS---------------
problem_type          fhtc
coordinate_frame      cartesian
transient|steady      transient
Nu                    12
Np                    10
mesh_file             cavity.neu
geometry_file         cavity.igs

-----------SOLVER PARAMETERS-------------
NR_iterations         10
linear_solver         bicgstab
linear_iterations     2000
restart_gmres         2000
left_precond          ilu0
right_precond         ilu0
tolerance             1e-12
initialization        value
UVPT_init_values      0.0  0.0  0.0  0.0  0.0  0.0
----------transient parameters-----------
numberoftimesteps     100
time_step_size        0.1
order                 3
num_ts_bef_output     1

-----------FLUID PROPERTIES--------------
viscosity             1.0
density               1.0
conductivity          1.0
specific_heat         1.0
--------buoyancy----------
buoyancy&direction    off  x
boussinesq            off
ref_density           1.0
ref_temperature       0.0
thermal_exp_coeff     .05
----------mhd-------------
mhdforce              off
Hartmann_Number       50.0

-----------SCALING PARAMETERS------------
Scaling_type          viscous
Length_Scale          1.0
Velocity_Scale        1.0
Temp_Diff_Scale       1.0
ref_temperature       0.0

----------BOUNDARY CONDITIONS------------
number_bnds           2
------------Flow Conditions--------------
boundary1             velocity  0.0  0.0
boundary2             velocity  if (t<1) -t else -1.0  0.0
------------Energy Conditions------------
boundary1             temperature  0.0
boundary2             temperature  0.0
```

Figure 3.4 : Input file for the solver.

The interpolation parameters are also included in this section through the velocity interpolation order *Nu*, pressure interpolation order *Np*, and the mesh file. Note that a geometry file name is given but the 3D code does not yet read in geometry files.

The second section is related to the solution of the matrix equations. In this section the user chooses the number of Newton-Raphson iterations, the linear solver with the number of iterations (and iterations before restart if *gmres* is used), the left and right preconditioners, the system residual tolerance, and the initialization type with values. The problem can be initialized with zeros, nonzero constants, variable definitions, or a file from a previous run. In the transient section, the typical time-stepping values such as the number of time steps, time step size, and order of the time integration are specified. The last entry sets the number of completed time steps before a data file is written.

The next section sets the fluid properties, whether or not buoyancy is present along with the direction in which it acts, and if buoyancy is present, whether or not the Boussinesq approximation is used. The Boussinesq approximation assumes that density is a function of temperature in the gravity terms, but is negligible for inertial terms and continuity. The density in the gravitational term then becomes

$$\rho = \rho_{ref}\left[1 - \beta(T - T_{ref})\right]$$

where ρ_{ref} is a reference density, T_{ref} is a reference temperature, and β is the thermal expansivity. If buoyancy is on, but the Boussinesq approximation is off, then density is considered a constant, resulting in a constant body force applied to the fluid. With

the Boussinesq approximation on, the reference values and the thermal expansivity must be set. The final portion of this section allows for an axial magnetic field to be applied in the full-zone problem. The magnetic field strength is determined by the Hartmann number.

Next is the scaling section. The scaling based on the viscous velocity scale used for the full-zone problem is currently the only option. The traditional viscous Reynolds number scaling seen in most fluid flow textbooks will be implemented in the near future.

The final section of the input file is the most important as it is largely what gives the code its generality. Here fluid and thermal boundary conditions are set for each boundary from a number of options. The types of fluid boundaries are *velocity*, *stress*, *mixed*$\cdots$, *velvar*$\cdots$, *strvar*$\cdots$, and *marangoni*. The conditions that contain the $\cdots$ require additional component-wise specifications. For instance, for a mixed condition where the x and y velocities and z normal stress component are specified, the dots become *vvs*. In each case except for *marangoni*, the (x, y, z) or (r, θ, z) components are given. In the case of the *marangoni* condition, the thermocapillary Reynolds number is given in the first entry and the other two are not read. The thermal conditions are included similarly with the options *temperature*, *flux*, *tempvar*, and *fluxvar* available.

The generality of the code is also increased through the use of variable initial and boundary conditions. The variable conditions are applied by using the "bench calculator" (*bc* command) provided in Unix. Within the code, an input file is created

containing a list of variables evaluated at the appropriate nodal values, which are then used in the variable expression. The input file is then inserted into *bc* via a system call, which creates an output file that is then read in by the code. While the system call and file read are "slow," they are insignificant compared to the amount of time required by other parts of the code such as solving the linear system. This provides a simple implementation by keeping all variable expressions contained in a single file.

There is one last note to be made here. The code does not account for units at all. The units used for fluid properties, input values, and boundary conditions are assumed to be made consistent by the user.

3.3 Constructing the Spectral Mesh

The first task for the code is to read in the input file and the mesh file provided by the user. This gives a list of elements and their eight corner nodes, but for the high-order interpolations employed here, it is necessary to construct a "spectral" mesh within each element. First, the one-dimensional flow and pressure grids must be obtained, from which a tensor product yields the full set of nodal points. The GLL and GL nodes defined in Section 2.3 and their corresponding quadrature weights are contained in a database constructed from MATLAB codes. However, these nodes are given on a cube defined by the interval $[-1, 1]$ in each direction and must be mapped to the physical hexahedral domain.

The mapping from the local element to the physical element in the code uses a linear blending proposed by Gordon and Hall [13, 14] and defined by

$$\begin{aligned}
\mathbf{x}(\varepsilon,\eta,\zeta) &= \frac{1-\varepsilon}{2}\pi_2(\eta,\zeta) + \frac{1+\varepsilon}{2}\pi_4(\eta,\zeta) + \frac{1-\eta}{2}\pi_5(\varepsilon,\zeta) \\
&+ \frac{1+\eta}{2}\pi_6(\varepsilon,\zeta) + \frac{1-\zeta}{2}\pi_3(\varepsilon,\eta) + \frac{1+\zeta}{2}\pi_1(\varepsilon,\eta) \\
&- \frac{(1-\varepsilon)(1-\eta)(1-\zeta)}{8}\mathbf{x}_4 - \frac{(1+\varepsilon)(1+\eta)(1-\zeta)}{8}\mathbf{x}_7 - \frac{(1-\varepsilon)(1-\eta)(1+\zeta)}{8}\mathbf{x}_2 \\
&- \frac{(1+\varepsilon)(1+\eta)(1+\zeta)}{8}\mathbf{x}_5 - \frac{(1+\varepsilon)(1-\eta)(1-\zeta)}{8}\mathbf{x}_3 - \frac{(1-\varepsilon)(1+\eta)(1-\zeta)}{8}\mathbf{x}_8 \\
&- \frac{(1+\varepsilon)(1-\eta)(1+\zeta)}{8}\mathbf{x}_1 - \frac{(1-\varepsilon)(1+\eta)(1+\zeta)}{8}\mathbf{x}_6
\end{aligned} \tag{3.1}$$

where ϵ, η, and ζ are the local coordinates, $\mathbf{x}_i$ are the nodal coordinates, and π_i are the face parametrizations. Currently, the face parameterizations are based on two-dimensional Gordon-Hall linear blending with all edges taking on a linear parametrization, resulting in a mapping that is simply a function of the nodal coordinates. Future versions of the code will allow for curved boundaries. However, it should be noted that in cylindrical coordinates, the boundary is represented exactly since the spatial coordinates are given by (r,θ,z) and the radius is constant.

In addition to building the spectral nodal set, it is necessary to do some book-keeping for the spectral mesh. New connectivity and boundary arrays are built to include all of the nodes.

3.4 Building Matrices

The next step is to compute the elemental matrices and assemble them into global matrices. All of the matrices that need to be built are contained in the matrix equations (2.14)–(2.19). Every entry of an elemental matrix is found by integrating over the elemental domain using Gaussian quadrature rules. The one-dimensional quadrature rule on the local interval is computed as

$$\int_{-1}^{1} f(x)dx = \sum_{q=1}^{N_q} f(x_q)w_q.$$

where w_q represents the quadrature weights, N_q is the order of the quadrature rule, and x_q are the quadrature points. If $N_q \geq 2N-1$, $f(x)$ is a polynomial of order N, and the GL nodes are used, the integration is exact. In general, the three-dimensional quadrature is analogous to the one-dimensional version, but in this case the tensor product definition of the basis functions means the integrand is separable so the integration is performed using one-dimensional quadrature rules in each direction. Also, since the elements are not on the local cube, the integral is mapped from the local to the physical element. This is accomplished through the Jacobian, and the complete integration takes the form

$$\int_{\Omega_e} f(r)g(\theta)h(z)d\Omega_e = \sum_{i=1}^{N_q}\sum_{j=1}^{N_q}\sum_{k=1}^{N_q} f(\varepsilon_i)g(\eta_j)h(\zeta_k)\left|J(\varepsilon_i,\eta_j,\zeta_k)\right| w_i w_j w_k$$

where each w is a set of one-dimensional quadrature weights and the Jacobian J is

$$J = \frac{\partial \mathbf{x}}{\partial(\varepsilon, \eta, \zeta)} = \begin{pmatrix} \frac{\partial r}{\partial \varepsilon} & \frac{\partial r}{\partial \eta} & \frac{\partial r}{\partial \zeta} \\ \frac{\partial \theta}{\partial \varepsilon} & \frac{\partial \theta}{\partial \eta} & \frac{\partial \theta}{\partial \zeta} \\ \frac{\partial z}{\partial \varepsilon} & \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \zeta} \end{pmatrix}.$$

Note that the vector quantity $\mathbf{x}$ used to compute the Jacobian is taken from the mapping in equation (3.1). Also, when the integrand contains a term with a spatial derivative such as $\frac{\partial f}{\partial r}$, the chain rule must be used such that $\frac{\partial f}{\partial x} = \frac{\partial f}{\partial \varepsilon}\frac{\partial \varepsilon}{\partial x} + \frac{\partial f}{\partial \eta}\frac{\partial \eta}{\partial x} + \frac{\partial f}{\partial \zeta}\frac{\partial \zeta}{\partial x}$. The partial derivatives of the local coordinates with respect to the physical coordinates (the geometric factors) are simply elements of the inverse Jacobian.

Now these tools are utilized to build the matrices. First, notice that the Lagrange polynomials and their derivatives with respect to the local variables on both the flow and pressure grids are computed on the same set of quadrature points for each element, so these values can be stored beforehand. The same weights are used as well so they too can be stored. The determinant of the Jacobian and the geometric factors change from element to element and thus must be computed and stored for each element.

A global matrix is built by looping over the elements, computing the entries of the elemental matrix via Gaussian quadrature, and "scattering" the elemental matrix to the global matrix. In the implementation, all entries are not scattered to the same global matrix; instead, a matrix corresponding to the unknowns and another corresponding to the Dirichlet conditions are built. The matrix information is stored

in two vectors: one with the entries and the other with a location given by $loc = n_{col}(row - 1) + col$ where n_{col} is the number of columns in the matrix and row and col give the entry's position. Once all elemental contributions have been added, the location vector is sorted and the matrix is converted to compressed sparse row (CSR) format.

3.5 Solving the Matrix System

Currently, the only method for solving the matrix system is by solving the fully-coupled equations all at once. Let the steady version of the nonlinear equations (2.14)–(2.19) take the concise form

$$\mathcal{F}(X) = B,$$

where X contains all of the unknowns. The nonlinearity of the operator $\mathcal{F}$ necessitates an iterative solution process. The Newton-Raphson method is chosen with the typical step

$$\mathcal{J}_k \Delta X = -R_k, \quad X_{k+1} = X_k + \Delta X$$

where

$$\mathcal{J}_k = \left.\frac{\partial \mathcal{F}}{\partial X}\right|_{X=X_k}$$

is the Jacobian matrix and

$$R_k = \mathcal{F}(X_k) - B$$

is the residual vector at the kth iteration. Thus the system of equations needs to be differentiated. This is a simple task for the linear terms, but the nonlinear terms must

be handled with greater care. Consider the convective term $\mathcal{C} = \tilde{U}_z \cdot \mathbf{C_{rz}} \tilde{U}_r$ found in the radial momentum equation. The derivatives of $\mathcal{C}$ with respect to U_r and U_z are needed and have the values

$$\frac{\partial \mathcal{C}}{\partial U_z} = \mathcal{I}_z \mathbf{C_{rz}} \tilde{U}_r, \quad \text{and} \quad \frac{\partial \mathcal{C}}{\partial U_r} = \tilde{U}_z \cdot \mathbf{C_{rz}}$$

where $\mathcal{I}_z$ represents a matrix that has ones on the diagonal for columns corresponding to unknowns and zeros for those corresponding to Dirichlet values. Taking derivatives of all of the equations gives the Jacobian matrix in its full form

$$\mathcal{J}_k = \begin{pmatrix} \mathbf{J_{rr}} & \mathbf{J_{r\theta}} & \mathbf{J_{rz}} & -\mathbf{G_r} & \mathbf{0} & \mathbf{D_{r\phi}} \\ \mathbf{J_{\theta r}} & \mathbf{J_{\theta\theta}} & \mathbf{J_{\theta z}} & -\mathbf{G_\theta} & Re_{tc}\mathbf{M_{B\theta}} & \mathbf{D_{\theta\phi}} \\ \mathbf{J_{zr}} & \mathbf{J_{z\theta}} & \mathbf{J_{zz}} & -\mathbf{G_z} & Re_{tc}\mathbf{M_{Bz}} & \mathbf{0} \\ \mathbf{G_r}^T & \mathbf{G_\theta}^T & \mathbf{G_z}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathcal{I}_r \mathbf{C_{Tr}} \tilde{T}^k & \mathcal{I}_\theta \mathbf{C_{T\theta}} \tilde{T}^k & \mathcal{I}_z \mathbf{C_{Tz}} \tilde{T}^k & \mathbf{0} & \mathbf{K_T} + \mathbf{N_T}^k & \mathbf{0} \\ \mathbf{D_{r\phi}}^T & \mathbf{D_{\theta\phi}}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{K}_\phi \end{pmatrix} \tag{3.2}$$

where

$$\mathbf{J_{rr}} = \mathcal{I}_r \mathbf{C_{rr}} \tilde{U}_r^k + \mathbf{N_r}^k + Ha^2 \mathbf{M_r} + \mathbf{K_{rr}}, \quad \mathbf{J_{r\theta}} = \mathcal{I}_\theta \mathbf{C_{r\theta}} \tilde{U}_r^k - \mathcal{I}_\theta \mathbf{C}_{\mathbf{r}\theta}^{\theta} \tilde{U}_\theta^k - \tilde{U}_\theta^k \cdot \mathbf{C}_{\mathbf{r}\theta}^{\theta} + \mathbf{K_{r\theta}},$$

$$\mathbf{J_{rz}} = \mathcal{I}_z \mathbf{C_{rz}} \tilde{U}_r^k + \mathbf{K_{rz}}, \quad \mathbf{J_{\theta r}} = \mathcal{I}_r \mathbf{C_{\theta r}} \tilde{U}_\theta^k + \tilde{U}_\theta^k \cdot \mathbf{C}_{\theta\theta}^{\mathbf{r}} + \mathbf{K_{\theta r}}$$

$$\mathbf{J_{\theta\theta}} = \mathbf{N}_\theta^k + \mathcal{I}_\theta \mathbf{C_{\theta\theta}} \tilde{U}_\theta^k + \mathcal{I}_\theta \mathbf{C}_{\theta\theta}^{\mathbf{r}} \tilde{U}_r^k + Ha^2 \mathbf{M}_\theta + \mathbf{K_{\theta\theta}}, \quad \mathbf{J_{\theta z}} = \mathcal{I}_z \mathbf{C_{\theta z}} \tilde{U}_\theta^k + \mathbf{K_{\theta z}},$$

$$\mathbf{J_{zr}} = \mathcal{I}_r \mathbf{C_{zr}} \tilde{U}_z^k + \mathbf{K_{zr}}, \quad \mathbf{J_{z\theta}} = \mathcal{I}_\theta \mathbf{C_{z\theta}} \tilde{U}_z^k + \mathbf{K_{z\theta}}, \quad \mathbf{J_{zz}} = \mathbf{N_z}^k + \mathcal{I}_z \mathbf{C_{zz}} \tilde{U}_z^k + \mathbf{K_{zz}}$$

and

$$\mathbf{N_i}^k = \tilde{U}_r^k \cdot \mathbf{C_{ir}} + \tilde{U}_\theta^k \cdot \mathbf{C_{i\theta}} + \tilde{U}_z^k \cdot \mathbf{C_{iz}}.$$

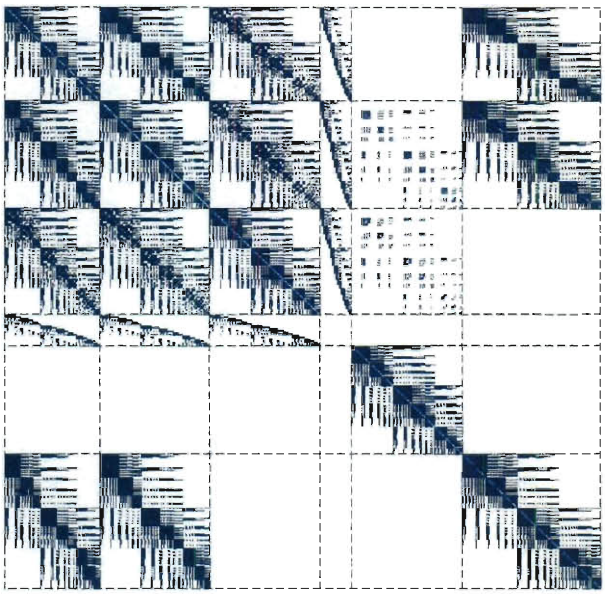


Figure 3.5 : A MATLAB spy plot (with color) of a Jacobian matrix built for the full-zone. The dashed lines form blocks that correspond to the entries in the Jacobian.

Again, in actually building this Jacobian matrix in the solver, the matrices are split into submatrices that correspond to unknowns and Dirichlet conditions. Only the matrices corresponding to the unknowns are included in the Jacobian matrix. The Dirichlet submatrices are used in computing the residual.

The structure of a Jacobian matrix for the full-zone problem can be seen in Figure 3.5. It is broken into blocks corresponding to (3.2).

3.5.1 Solving the Linear System

The majority of the solver's time is typically spent in solving the linear system, so it is imperative that the system is solved efficiently. Currently, the code has

two linear solvers from which to choose: gmres and BiCGStab. BiCGStab tends to stagnate when solving complex flows so gmres is the primary linear solver used here. Each of these methods are based on projecting the solution into a Krylov subspace, which is much smaller than the original space.

However, the Krylov subspace methods alone can be incapable of solving the system to the desired tolerance, especially for complex flows. To aid the linear solver, preconditioners are used to lower the condition number of the matrix and make the problem easier to solve. The code has three options for preconditioners: diagonal, block diagonal, and ILU(0). The diagonal preconditioner simply uses the diagonal of the matrix as a preconditioner, the block diagonal preconditioner picks out uncoupled blocks on the diagonal that correspond to each element, and the ILU(0) preconditioner performs an incomplete LU decomposition with zero fill, meaning that the nonzero locations in the system matrix are the same for $L + U$.

Much like in the case with choosing a linear solver, there is only one option that proves to provide sufficient preconditioning – ILU(0). While the diagonal does offer some speed-up (by reducing the total number of iterations it takes to converge), it is inadequate for flows at high Reynolds numbers. The block Jacobi preconditioner works for much higher Reynolds numbers than the diagonal preconditioner, but still suffers from inadequacy for Reynolds numbers needed for the simulations of interest. The ILU(0) preconditioner provides much more speed-up than the previous two preconditioners, but takes longer to build and uses more storage. However, these

Linear Solver Comparison: Average Iterations				
	None	Diagonal	Block Diagonal	ILU(0)
GMRES	2291	2023	1094	99
BiCGStab	-	-	6205	111

Table 3.1 : Iterations, averaged over 7 Newton-Raphson steps, needed in solving the half-zone liquid bridge problem with $Re_{tc} = 1500$. Dashes represent divergence.

disadvantages are tolerated so that larger Reynolds number flows can be computed.

The algorithms for the preconditioned gmres and the preconditioned BiCGStab routines, as well as the algorithm for building the ILU(0) preconditioner can be found in Appendix A.

Tables 3.1 and Table 3.2 give a better idea of how the preconditioners affect each linear solver. The tables were created by simulating the steady-state flow in the half-zone liquid bridge with $Pr = 0.01$ and $Re_{tc} = 1,500$ (see Section 4.4). The simulations were initialized with the solution from $Re_{tc} = 1,000$ and performed on a $5 \times 5 \times 5$ grid with 3rd and 1st order interpolation functions for the flow and pressure grids, respectively. The two tables give the average number of linear iterations for the steps in the Newton-Raphson method (7 steps in this case) and the total time needed to build the preconditioners and solve the linear systems. The tables show that gmres with the ILU(0) preconditioner are the best combination.

Linear Solver Comparison: Time (min)				
	None	Diagonal	Block Diagonal	ILU(0)
GMRES	51.48	45.34	22.19	2.26
BiCGStab	-	-	51.13	3.36

Table 3.2 : Total time elapsed while building preconditioners and solving the linear systems for the half-zone liquid bridge problem with $Re_{tc} = 1500$. Dashes represent divergence.

3.5.2 Transient Solutions

Next, let the time-dependent terms return to the equations and write them as

$$\mathbf{M}\frac{\partial X}{\partial t} + \mathcal{F}(X) = B. \tag{3.3}$$

Equation (3.3) is an ordinary differential equation, which can be solved by using one of the many numerical time integration techniques. A Backwards Differentiation Formula (BDF)

$$\frac{dX}{dt} = \frac{\sum_{q=0}^{N_t} \beta_q X^{n-q}}{\Delta t}$$

is used to represent the derivative and this leads to an iterative time-stepping method. Here, N_t is the order of time integration, Δt is the time step size, n is the current time step, and X^{n-q} is the value of X at time step $n-q$. Substituting this into (3.3) gives

$$\mathbf{M}\frac{\sum_{q=0}^{N_t} \beta_q X^{n-q}}{\Delta t} + \mathcal{F}(X) = B$$

$$\Rightarrow \quad \beta_0 \mathbf{M}\frac{X^n}{\Delta t} + \mathcal{F}(X^n) = B - \mathbf{M}\frac{\sum_{q=1}^{N_t} \beta_q X^{n-q}}{\Delta t}.$$

order	β_0	β_1	β_2	β_3	β_4	β_5	β_6
1	1	-1					
2	3/2	-2	1/2				
3	11/6	-3	3/2	-1/3			
4	25/12	-4	3	-4/3	1/4		
5	137/60	-5	5	-10/3	5/4	-1/5	
6	49/20	-6	15/2	-20/3	15/4	-6/5	1/6

Table 3.3 : Weights for the Backwards Differentiation Formula.

As in the steady case, this equation is nonlinear and must be solved with the Newton-Raphson method. Thus, at time step n, the iteration is

$$\left[\beta_0 \frac{\mathbf{M}}{\Delta t} + \mathcal{J}(X_k^n)\right] \Delta X^n = -R_k^n, \tag{3.4}$$

where the residual vector at the nonlinear iteration k is

$$R_k^n = \mathbf{M}\frac{\beta_0 X_k^n + \sum_{q=1}^{N_t} \beta_q X^{n-q}}{\Delta t} + \mathcal{F}(X_k^n) - B = \mathbf{M}\frac{\beta_0 X_k^n + \sum_{q=1}^{N_t} \beta_q X^{n-q}}{\Delta t} + R_{k,ss}^n.$$

The steady state residual vector is represented by $R_{k,ss}^n$. The code contains the weights for this time-stepping method up to order six. The weights are listed in Table 3.3.

The implementation of this time-stepping method is straightforward. The Jacobian matrix is simply updated with mass matrices and N_t previous solution vectors are stored for use on the RHS. However, there is one important implementation detail to note here. For the first time step, the only previous "solution" available is the ini-

tial condition, so for this step the method degenerates to a first-order method. This lack of previous solution vectors means the BDF order ramps until it reaches the order specified by the user. The effect this has on the accuracy of the time integrations for all orders is shown in Section 4.3.

Chapter 4

Benchmarking

In this chapter the solver's spatial and temporal accuracy is validated by applying it to four benchmark problems. The results obtained by the solver are compared with either an exact solution or a numerical solution which has been verified. The first problem is the 3D lid-driven cavity flow, which provides benchmarks quantitatively based on previous numerical results and qualitatively based on physical intuition. The second problem, Kovasznay flow, has an exact three-dimensional solution in cylindrical coordinates so it benchmarks the cylindrical portion of the solver. The third flow is the Taylor vortex, which has an exact transient solution in Cartesian coordinates, and is used to benchmark the temporal accuracy. The final problem is the half-zone model of the thermocapillary driven liquid bridge. This provides benchmarks for the thermocapillary conditions and validates the mesh that is used to obtain full-zone results in the next chapter.

All velocity vectors, contours, and streamline plots are made through the use of the open-source visualization software Paraview [15].

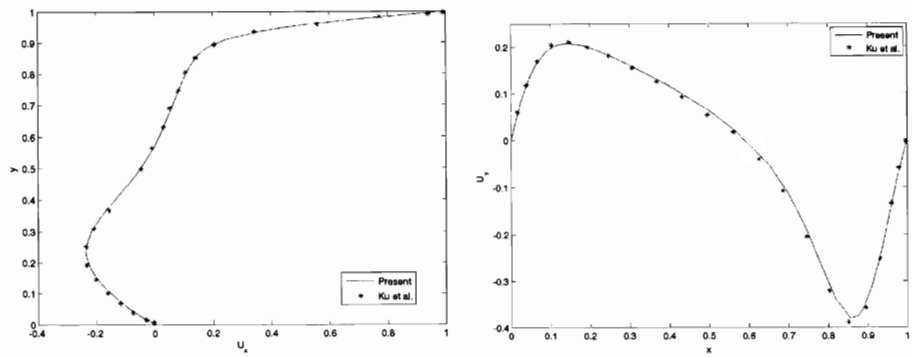


Figure 4.1 : Lid driven cavity velocity profiles for $Re = 400$: (a) x (horizontal) velocity component along vertical centerline; (b) y (vertical) velocity component along a horizontal centerline.

4.1 Lid-Driven Cavity

The lid driven cavity is a typical benchmark problem for Cartesian coordinates. The unit cube serves as the domain with the lid represented by the side at $y = 1$. The boundary conditions are $\mathbf{u} = (1, 0, 0)$ at $y = 1$ and $\mathbf{u} = \mathbf{0}$ on all other boundaries. The Navier-Stokes equations govern the flow field and the only parameter in the nondimensionalized system is the viscous Reynolds number Re. No exact solution exists for this problem, but it has been studied numerically [16, 17].

For this simulation, a uniformly structured grid with ten elements in each direction and fourth-order interpolations was used. The Reynolds number was set to $Re = 400$ to compare with the results from Ku et al. [16] (see Figure 4.1), who used a pseudospectral matrix method, and Guj and Stella [17], who used a second-order finite difference scheme on a grid with $101 \times 101 \times 81$ nodes. As shown in Figure 4.1, the results from the solver agree well with previous results. Additionally, Figure 4.1 shows

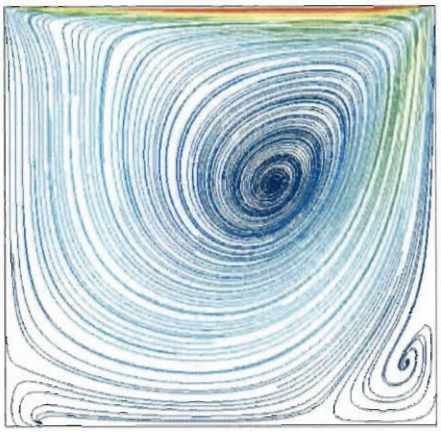


Figure 4.2 : Streamlines for the lid driven cavity flow projected onto the $z = 0.5$ plane.

the streamlines for the flow projected onto the $z = 0.5$ plane. The typical upstream circulation can be seen in the back-bottom corner and the larger downstream vortex exists as expected in the front-bottom corner.

4.2 Kovasznay Flow

The next test case is the three-dimensional Kovasznay flow in a cylindrical domain. In 1948, L.G. Kovasznay found a solution for flow behind a two-dimensional grid [18] and recently, Blackburn and Sherwin extended this solution to exact three-dimensional flow in a cylindrical domain [19]. Note that this flow is truly three-dimensional as the azimuthal symmetry can be broken by off-axis placement of the

grid. The exact solution is given by

$$u_r = (2\pi)^{-1} \lambda \exp(\lambda z) \sin(2\pi [r\cos(\theta + \Theta) + \Delta]) \cos(\theta + \Theta)$$

$$u_\theta = -(2\pi)^{-1} \lambda \exp(\lambda z) \sin(2\pi [r\cos(\theta + \Theta) + \Delta]) \sin(\theta + \Theta)$$

$$u_z = 1 - \exp(\lambda z) \cos(2\pi [r\cos(\theta + \Theta) + \Delta])$$

$$p = (1 - \exp(2\lambda z))/2$$

where $\lambda = Re/2 - (Re^2/4 + 4\pi^2)^{1/2}$ and Re is the viscous Reynolds number, Θ is a rotational offset, and Δ is a translational offset from the axis. Note that $\Delta \neq 0$ yields flow that crosses the axis.

For this case, $Re = 40$, $\Delta = 0.1$, and $\Theta = 0.5$ were chosen. A mesh with 3240 elements evenly spaced in each direction with third-order velocity basis functions was used. The simulation was run until the root-mean-square (rms) residual was 10^{-12}. The error at the nodal points for the axial velocity was 7.53×10^{-4} in the max norm and 4.67×10^{-7} in the rms norm. These error norms confirm the ability to solve three-dimensional cylindrical flows. Figure 4.3 shows the velocity vectors on the plane defined by $\theta = \Theta = 0.5$ *rad*.

4.3 Taylor Vortex

To test the accuracy of the time-stepping scheme using Backwards Differentiation Formulae (BDF), a Taylor vortex is simulated as the time-dependent solution has the

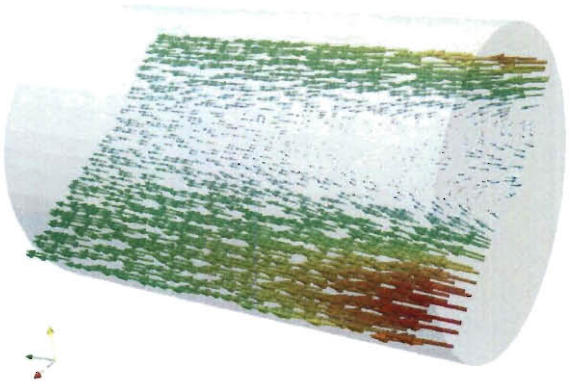


Figure 4.3 : Velocity vectors on the plane $\theta = \Theta$ for the cylindrical Kovasznay flow.

exact form

$$u_x = -\cos(x)\sin(y)e^{-2t/Re}$$

$$u_y = \sin(x)\cos(y)e^{-2t/Re}$$

$$p = -\frac{1}{4}\left[\cos(2x) + \cos(2y)\right]e^{-4t/Re}$$

where Re is the viscous Reynolds number. As the exact solution reveals, the Taylor vortex is a two-dimensional flow in Cartesian coordinates. It serves as a viable transient benchmark, as the spatial correctness of the code has already been established.

For the three-dimensional simulation, a hexahedral domain defined by the intervals $[-\pi/2, \pi/2]$ in x and y and $[0, 1]$ in z was used. All boundary conditions for u_x and u_y and were time-dependent Dirichlet conditions extracted from the exact solution, and homogeneous boundary conditions were used for u_z. For the simulation $Re = 1$ was set and time integations of orders one through six were carried out for 20 time steps, with time-step size $\Delta t = 0.05$. Then, the simulation was run a second time

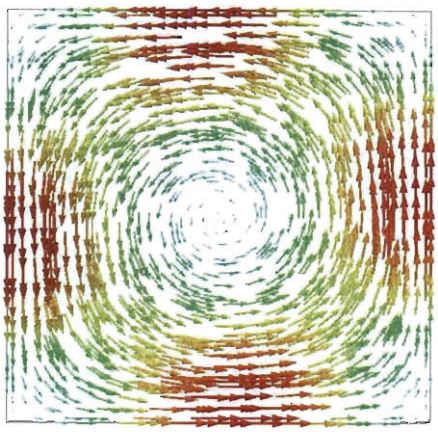


Figure 4.4 : The taylor vortex velocity vectors.

with 40 time steps with $\Delta t = 0.025$. The 2-norm of the error in the nodal values is plotted against the order of time integration in Figure 4.3 for both cases.

The figure shows that little additional accuracy is gained by usign BDF orders higher than 3rd order. We believe this is due to the initial ramping of orders that moves the time-stepping from 1st order in the first time step, to higher orders in consecutive time steps. The error in the initial low order time steps and its subsequent propagation set an error that can not be quickly overcome by simply raising the order of integration. Thus, with the current implementation, there is no need to use time integration orders greater than three.

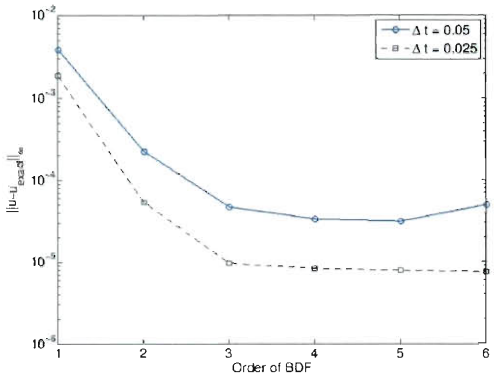


Figure 4.5 : Error of the Taylor vortex simulation at nondimensional time $t = 1.0$ against the order of time integration.

4.4 Half-Zone Model of the Liquid Bridge

The final benchmark problem was described briefly in Chapter 1 and is quite similar to the full-zone problem statement given in Chapter 2. The difference in the two models is in the portion of the melt-zone treated, which affects the boundary conditions that are applied. For the half-zone, the boundary conditions

$$\mathbf{u} = \mathbf{0},\ T = 1 \quad \text{at} \quad z = 0$$

$$\mathbf{u} = \mathbf{0},\ T = 0 \quad \text{at} \quad z = 1$$

$$\sigma_{rz} = -Re_{tc}\frac{\partial T}{\partial z},\ \sigma_{r\theta} = -Re_{tc}\frac{1}{r}\frac{\partial T}{\partial \theta},\ \frac{\partial T}{\partial r} = 0 \quad \text{at} \quad r = 1$$

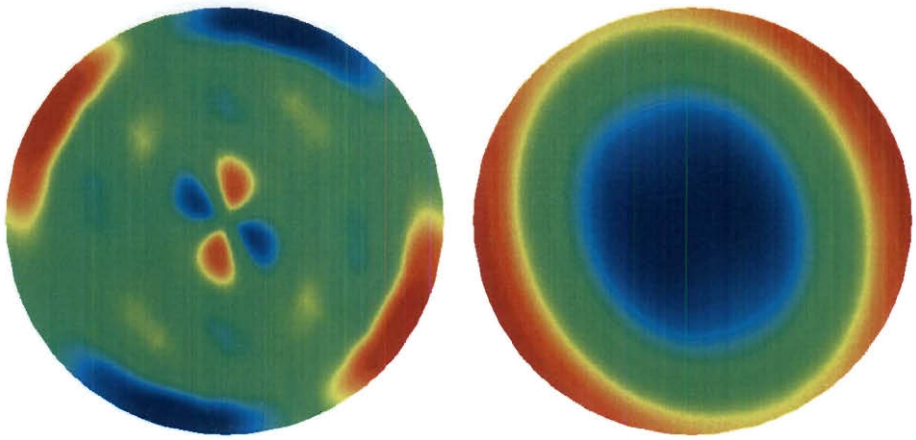
are applied where Re_{tc} is the thermocapillary Reynolds number given in Chapter 2.

Here, the terminology used to identify perturbation modes is introduced. In the normal-mode linear stability analysis by Huang and Houchens [10], m is the azimuthal wave number used in the expression for perturbations to the axisymmetric base state.

The value of m must be an integer because the wave must be continuous (single valued) at $\theta = 0\ rad$ and $\theta = 2\pi\ rad$, since they represent the same point in space. Physically, the wave number refers to the number of spatial periods in the azimuthal direction, which creates $2m$ wedges of flow because of the peak and trough in each period. An example of this can be seen in Figure 4.6(a). Also, the modes are described as either stationary (or steady) or periodic corresponding to time-indepedent and time-dependent modes, respectively.

For the half-zone problem with $Pr = 0.01$, the base state is axisymmetric until the initial instability occurs at approximately $Re_{tc} = 1960$, which results in a $m = 2$ stationary mode. Levenstam and Amberg [4] performed three-dimensional simulations for the steady, three-dimensional mode from the critical Re_{tc} up to the point of secondary instability, a periodic $m = 2$ mode, which occurred at approximately $Re_{tc} = 6250$. Then they continued to simulate the periodic, three-dimensional modes up to $Re_{tc} = 10,000$. Here, the goal is to verify that the same modes and maximum velocities are found for the same Re_{tc} values.

The simulations in the present study were run on four different 3rd order meshes. For $Re_{tc} = 3500$, Figure 4.6 shows the azimuthal velocity and the temperature plotted at $z = 0.5$. It is easy to see that four cells (wedges) appear, corresponding to the mode $m = 2$ as expected. Blue regions in the azimuthal velocity indicate flow in the counter-clockwise direction, and red indicates clockwise flow. There are noticeable alternating "cold" (blue) and "hot" (red) spots in the temperature field that coincide

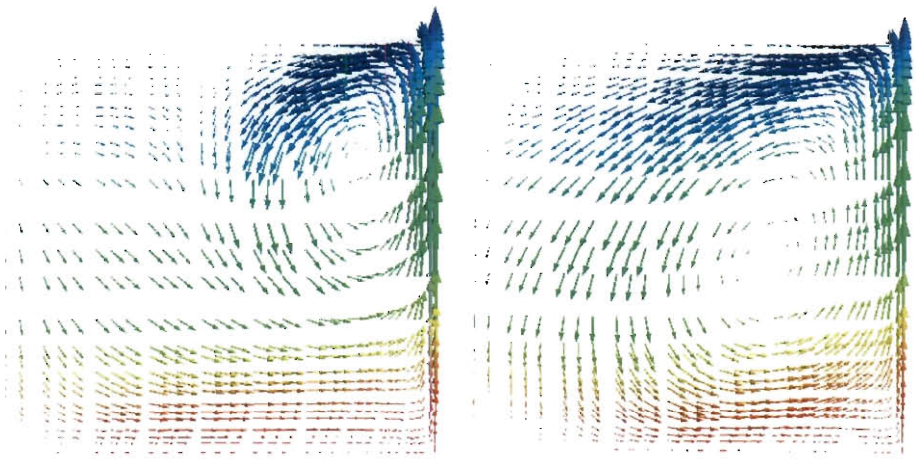


(a) (b)

Figure 4.6 : Plots on the $z = 0.5$ plane in the half-zone of a) the azimuthal velocity and b) the temperature.

with the cell interfaces. The azimuthal velocity plot suggests that these form because the flow is carrying thermal energy from the cold spots to the hot spots. Figure 4.7 shows the difference in the flow fields for a plane bisecting the hot region and a plane bisecting the cold region.

All figures agree well with the results in [4]. Table 4.1 compares the values of the maximum velocity magnitude and the azimuthal velocity component to the results of Levenstam and Amberg [4] for different mesh sizes. In the present study, the coarsest mesh underpredicts the azimuthal velocity by nearly 20% and while azimuthal refinement helps, it is the additional refinement in the radial and axial directions that gives the expected value for the azimuthal velocity. A comparison of the bottom row of the table shows that the azimuthal velocities match up exactly and the difference



(a) (b)

Figure 4.7 : Velocity vectors on a plane that bisects a) the hot spot and b) the cold spot.

Levenstam and Amberg [4]			Present Study		
Grid	U_{max}	$U_{\theta,max}$	Grid	U_{max}	$U_{\theta,max}$
$10 \times 16 \times 10$	0.08876	0.0115	$31 \times 24 \times 31$	0.08740	0.0110
$20 \times 16 \times 20$	0.08872	0.0130	$31 \times 48 \times 31$	0.08677	0.0119
$20 \times 32 \times 20$	0.08671	0.0132	$31 \times 96 \times 31$	0.08677	0.0115
$30 \times 48 \times 30$	0.08647	0.0134	$43 \times 48 \times 43$	0.08673	0.0134

Table 4.1 : Maximum values of the velocity magnitude and the azimuthal velocity component at $Re = 3500$ for different grids, which are defined by the number of nodes in each direction ($N_r \times N_\theta \times N_z$).

in maximum velocities is only 0.3%.

Chapter 5

Full-Zone Results

In this chapter, the results for the simulations of the full-zone model with and without magnetic stabilization are presented. The results give the intervals in thermocapillary Reynolds number across which the initial instability is tripped. The modes are visualized and discussed, and the effects of magnetic stabilization are demonstrated.

In the following sections, the modes are described in the same way they were in Section 4.4. In this case, perturbations are divided into "symmetric" and "anti-symmetric" modes, which describe the symmetry of perturbation variables. The symmetry is most obviously demonstrated by comparison of the azimuthal velocity above and below the axial midplane. The Prandtl number is fixed at $Pr = 0.02$.

5.1 The Critical Point

For the full-zone model of optically heated float-zone crystal growth, the base state is a pair of axisymmetric toroidal flows that are also symmetric about the midplane. This state remains until the thermocapillary Reynolds number Re_{tc} reaches a sufficiently large critical value $Re_{tc,cr}$. When growing crystals, knowledge of this critical value allows for the magnitude of the heat flux (as this is the controllable

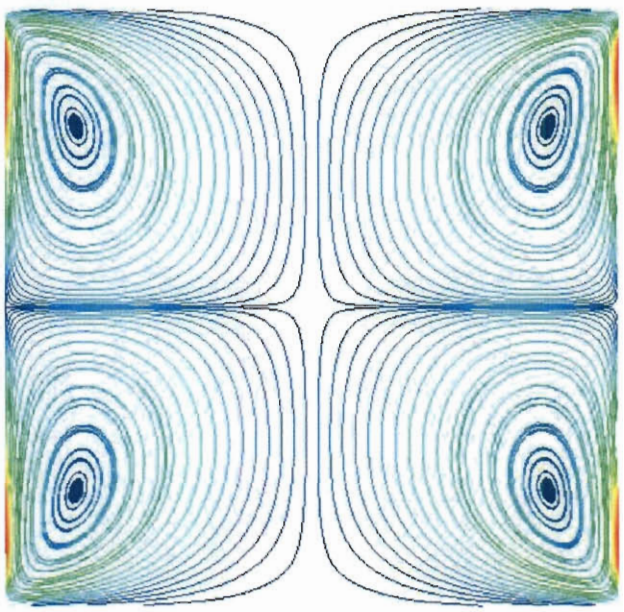


Figure 5.1 : The base flow streamlines on any $r - z$ plane in the cylinder. Note that the flow is symmetric about the $r = 0$ and $z = 0$ planes.

quantity in an experiment) to be selected sufficiently low so that $Re_{tc} < Re_{tc,cr}$ and nonaxisymmetric instabilities are avoided. However, a minimum Re_{tc} is also required to maintain a sufficient temperature gradient to avoid the morphological instability at the growth front, which results in polycrystalline growth. Because of this competition it is advantageous to stabilize the base state through the application of an axial magnetic field. Here the critical value $Re_{tc,cr}$ for Hartmann numbers ranging from $Ha = 0$ to $Ha = 10$ is investigated.

To track down the instabilities, the results from linear stability theory were used to predict near what values of Re_{tc} to search. Time-dependent simulations were run for several values of Re_{tc} and the maximum value of the azimuthal velocity v_θ was

Critical Re_{tc}			
	Linear Stability	3D Simulation Bounds	
Ha		stable	unstable ($V_{\theta,max}$)
0	1,548.56	1,600	1,650 (3.41)
5	1,842.97	1,900	1,950 (5.67)
10	2,629.52	2,650	2,750 (6.81)

Table 5.1 : A comparison of the critical points found in linear stability theory [3, 10] with the bounds found through 3D transient numerical simulations. The value of the maximum azimuthal velocity is given in parenthesis for the upper bound of the 3D simulations.

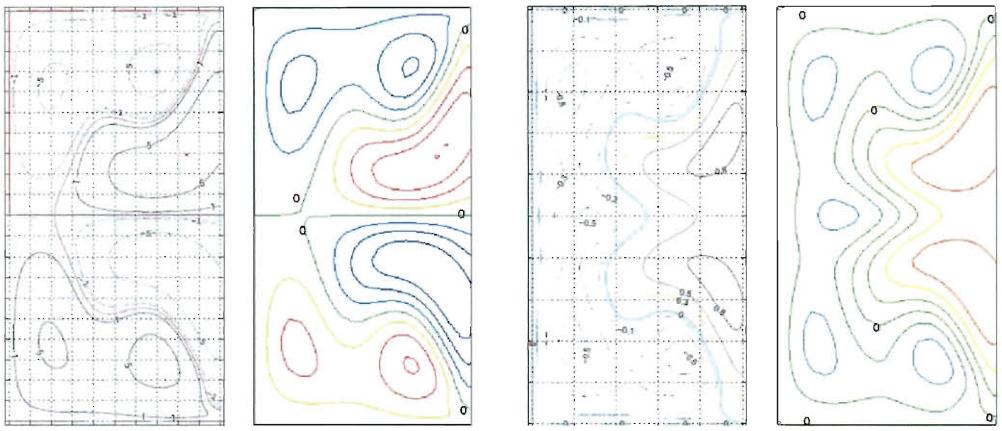
monitored. If the value of v_θ decayed and remained small, it was assumed that the critical value had not been reached. In contrast, growth in the azimuthal velocity indicates the instability was present. Pinpointing the critical point exactly using 3D numerical simulations is not an easy task as a simulation must be run for every candidate Re_{tc} value. Instead, lower and upper bounds were found and are given in Table 5.1. Also, the maximum azimuthal velocity found at the instability for the 3D simulations is given as it is expected to decrease as the thermocapillary Reynolds number decreases and we move closer to the point of instability. Overall, the 3D simulations predict $Re_{tc,cr}$ values that are consistently only $\sim 3 - 6\%$ higher than those found using linear stability theory. More importantly, the 3D simulations and the linear stability analyses show that stabilization via an axial magnetic field does in fact have the desired effect of increasing the critical thermocapillary Reynolds number

via stabilization of the base flow.

5.2 $Ha = 0$ Modes

In addition to finding the critical Re_{tc} values, the wave number and nature of the mode (stationary or periodic) predicted by linear stability theory are hoped to be matched by the simulations. In linear stability theory the critical value for each potential mode is found and the mode with the smallest critical value is expected to appear first. However, linear stability does not necessarily give any indication of what happens after the initial instability appears. The perturbation may attain nonlinear saturation and stay stable, or transition again to some possibly unknown state. Time-dependent 3D numerical simulations, on the other hand, can approximate the point of initial instability and continue computing the flow field after it becomes unstable.

Linear stability theory predicts that in the case with no applied magnetic field, a stationary anti-symmetric $m = 2$ mode is the first instability at a critical value of $Re_{tc} = 1546.58$ [10]. As seen in Section 4.4, this mode is also found in the half-zone model. For $Re_{tc} = 1650$, the $m = 2$ stationary anti-symmetric mode is observed in the present study. The linear stability analysis is not able to produce a 3D flow field at the instability for comparison with the simulations, but it does provide perturbation velocities for all velocity components. Since the base flow is axisymmetric, any azimuthal velocity is the result of perturbations, allowing the azimuthal velocity in the 3D simulations to be compared to the perturbations (eigenmodes) from the



(a) (b) (c) (d)

Figure 5.2 : The azimuthal velocity contours for a) the anti-symmetric mode from linear stability theory, b) the anti-symmetric mode from 3D simulations, c) the symmetric mode from linear stability analysis, and d) the symmetric mode from 3D simulations. The contours from the 3D simulations for the anti-symmetric mode are at $1.91n$ for $n = -3, \ldots, 3$ and for the symmetric modes, the contours are at $1.33n$ for $n = -2, \ldots, 3$.

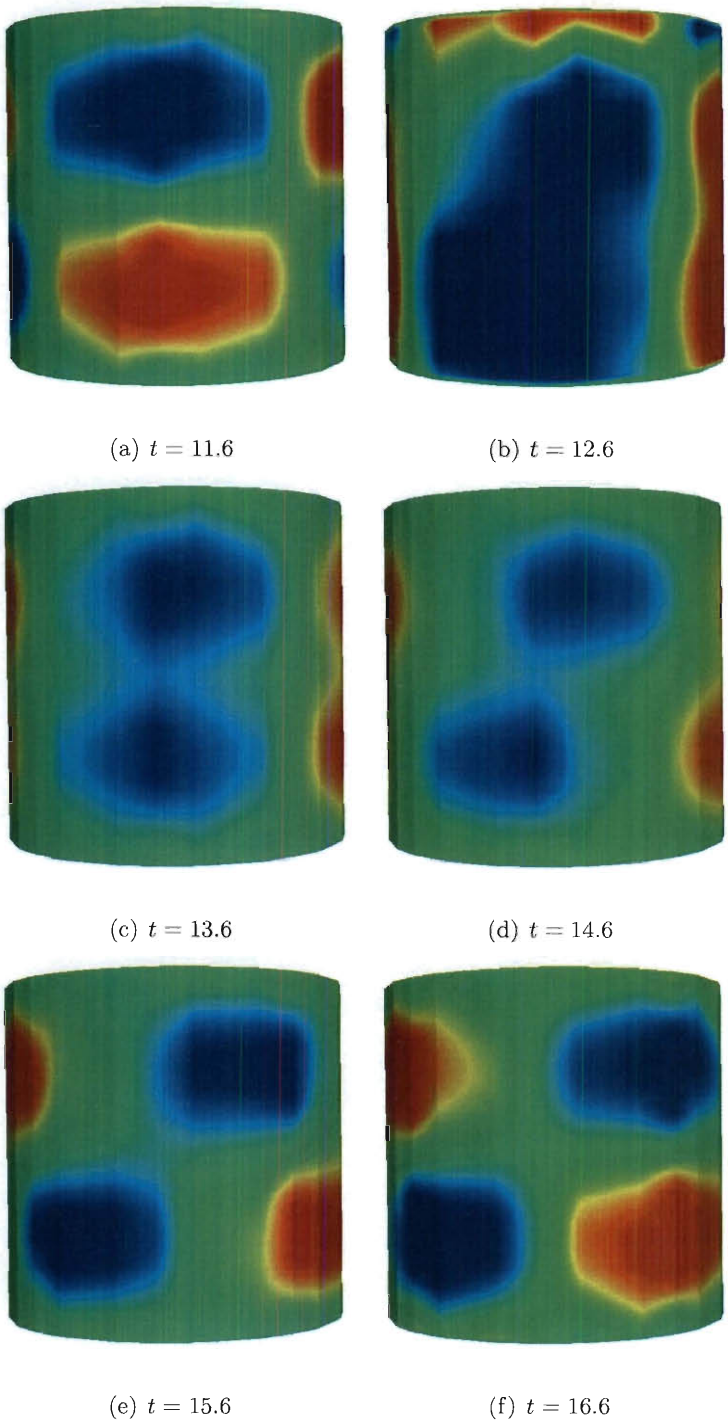
linear stability analysis. Figures 5.2(a) and 5.2(b) show the contours of the azimuthal velocity for the antisymmetric mode computed with linear stability theory and 3D simulations. As can be seen in the figure, there is very good agreement between the two.

Whereas the simulation at $Re_{tc} = 1650$ found the stationionary $m = 2$ mode as expected, the simulation at $Re_{tc} = 1700$ exhibits far different behavior. First, the anti-symmetric mode appears as expected, but it does not reach steady state as it does in the half-zone case or at $Re_{tc} = 1600$ in the full-zone. Instead, the anti-symmetric mode quickly disappears and the flow jumps to a symmetric $m = 2$ mode.

A comparison of this mode's azimuthal velocity from 3D simulations to the azimuthal velocity perturbations from linear stability analysis is given in Figures 5.2(c) and 5.2(d). From the symmetric state, the cells begin to migrate in opposite azimuthal directions with respect to the midplane. Each cell moves approximately 45° until the flow reaches the anti-symmetric mode again, where the cycle starts over. This process can be seen in Figure 5.3.

Figure 5.4 shows the maximum azimuthal velocity against time. Matching this plot up with Figure 5.3, it can been seen that the flow state in Figure 5.3(a) is just to the left of a peak in the azimuthal velocity. The peak corresponds to the next state where the flow is transitioning from an anti-symmetric to a symmetric state. From the plot, the period of the modal competition is not constant, but may become constant with further time integration. The times between the three peaks are 7.6, 5.6, and 5.9 nondimensional time units.

Next, the instability mechanism at $Re_{tc} = 1600$ is described by analyzing the flow field before, at, and after the transition point between the competing modes. In Figure 5.3(a), the anti-symmetric mode is present, meaning that the cells are counter-rotating with respect to the azimuthal direction. Then, in Figure 5.3(b), the azimuthal velocity is all in the same direction but there do not appear to be entirely separate cells. This can be explained by viewing the velocity vectors in a plane bisecting the cells at the transition point, which is shown in Figure 5.5. From the figure, it is apparent that the axial anti-symmetry is broken as the cell



(a) $t = 11.6$ (b) $t = 12.6$

(c) $t = 13.6$ (d) $t = 14.6$

(e) $t = 15.6$ (f) $t = 16.6$

Figure 5.3 : Time instances of the surface flow with $Ha = 0$ and $Re_{tc} = 1750$. The color represents the azimuthal velocity. The six instances cover the period of the of the cycle.

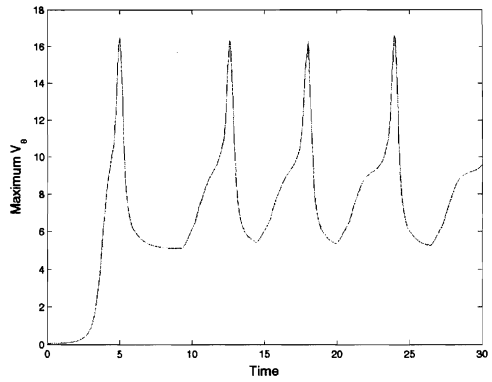


Figure 5.4 : Maximum azimuthal velocity for the $Ha = 0$, $Re_{tc} = 1750$ case.

on top is sagging into the cell on bottom. This behavior is expected as perturbation vectors from linear stability theory also indicate a breaking of antisymmetry [10]. This interaction of the cells eventually slows the rotation of the bottom cell and forces it to change its direction of rotation, yielding the symmetric mode.

The appearance of both modes is an interesting result because there is competition between the two steady modes that makes them appear to be a single periodic mode. Since the critical value $Re_{tc} = 1618.42$ [3] for the symmetric mode is not much higher than that of the anti-symmetric mode, the behavior should not be entirely surprising.

One consequence of breaking of the axial midplane symmetry has already been shown: this type of behavior can not be found in the half-zone. The half-zone model is based on the assumption that symmetry about the midplane is not broken, but that flow dynamics similar to the full-zone model exist. However, these results show that the half-zone model is inadequate for predicting the competition of modes observed

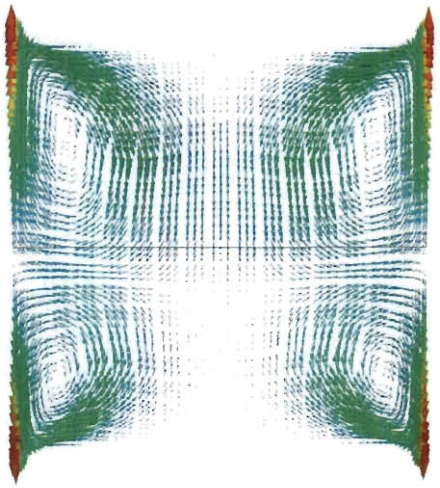


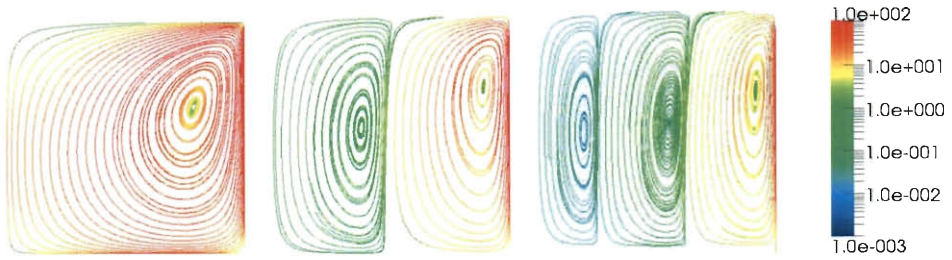
Figure 5.5 : The transition from the anti-symmetric $m = 2$ mode to the symmetric $m = 2$ mode occurs due to the break in symmetry across the axial midplane (represented by the black line).

in the full-zone.

5.3 Effects of Magnetic Stabilization

The reason for applying an axial magnetic field to the flow is to induce Lorentz body forces that oppose the flow in the radial and azimuthal directions. This is expected to damp perturbations such that more energy is required to force the onset of the instability. Table 5.1 in Section 5.1 shows that this is the case as the magnetic stabilization delays the onset of the initial instability.

In addition to damping the perturbations, the Lorentz force also opposes flow radially and azimuthally in the base flow, which keeps the flow near the free surface. This rotational flow causes a counter-rotating "cell" to form in the interior. As the Hart-

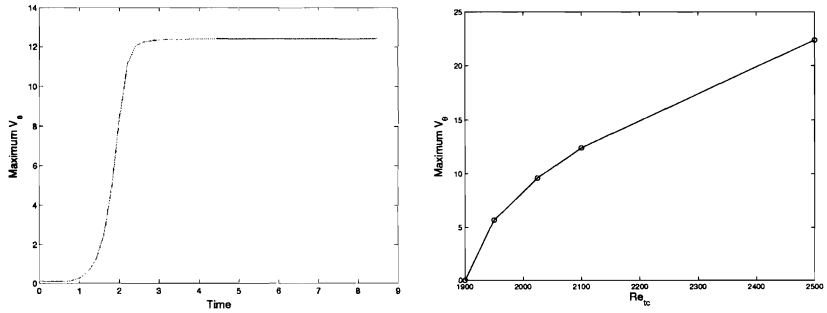


(a) (b) (c)

Figure 5.6 : Formation of cells in the base flow for a) $Ha = 0$, b) $Ha = 25$, and c) $Ha = 50$ all at $Re_{tc} = 1000$. The streamlines are colored by velocity magnitude on a logarithmic scale.

mann number grows, the Lorentz forces become stronger so the cells become thinner, which forces more cells to form in the interior. However, each new cell that appears is weaker than the previous. The development of the cells is shown in Figure 5.6 from which it can be seen that the velocity magnitude drops by approximately two orders of magnitude from cell to cell. This is another advantage of the magnetic stabilization because it leaves a fairly quiescent interior, which is desired in the float-zone growth process.

Simulations also suggest that magnetic stabilization changes the nature of the instability. Whereas the anti-symmetric and symmetric $m = 2$ modes alternated at high enough Re_{tc} for the case with no magnetic field, for the nonzero Hartmann numbers investigated here ($Ha = 5$ and $Ha = 10$), there was no modal competition. For all nonzero Hartmann number cases, the anti-symmetric mode formed and stabilized. Physically, the perturbations that caused the transition before are now

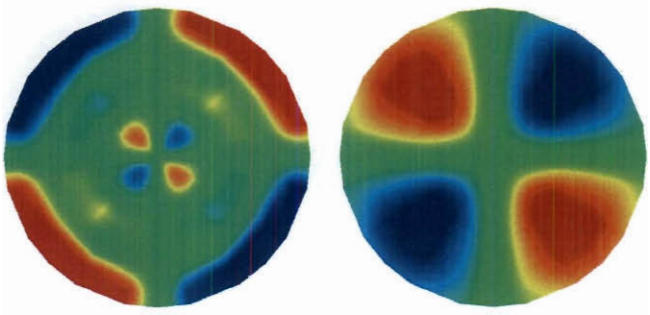


(a) (b)

Figure 5.7 : Maximum azimuthal velocity for $Ha = 5$ vs. a) nondimensional time with $Re_{tc} = 2100$ and b) thermocapillary Reynolds number.

being damped, leaving just the one dominant steady mode. The plot of the maximum azimuthal velocity against time in Figure 5.7(a) for $Ha = 5$ shows that the value stabilized unlike the value in Figure 5.4 for the $Ha = 0$ case. Figure 5.7(b) shows that value of the maximum azimuthal velocity initially increases quickly as the thermocapillary Reynolds number is increased beyond the critical point, but the increase becomes smaller as Re_{tc} grows. This nonlinear saturation behavior is similar to that seen in simulations of the half-zone [4].

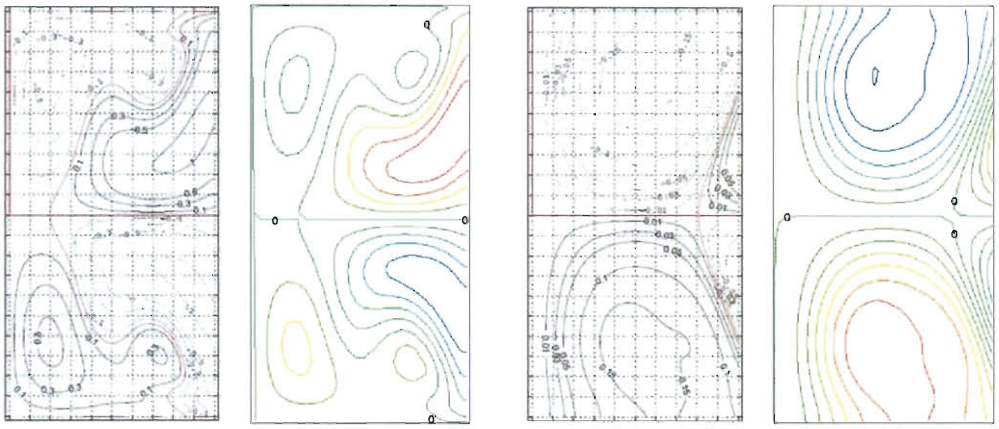
The modes for $Ha = 5$ and $Ha = 10$ were compared to linear stability analyses [10] just as in the $Ha = 0$ case. Figure 5.3 shows the appearance of the $m = 2$ mode by plotting the azimuthal velocity and the electric potential on $r-\theta$ planes. Figure 5.3 compares the contours of azimuthal velocity and electric potential at $Ha = 5$ to those from linear stability theory. Note that comparisons can be made with linear stability for the electric potential because, just like the azimuthal velocity, the electric potential



(a) (b)

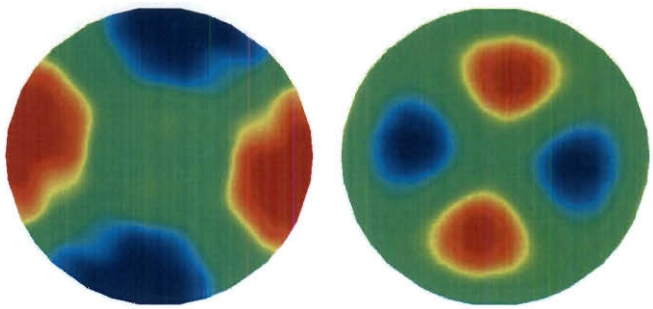
Figure 5.8 : The stationary $m = 2$ mode for $Ha = 5$ at $Re_{tc} = 1950$ shown in plots of a) the azimuthal velocity and b) electric potential.

is zero in the axisymmetric base state, which means that any electric potential seen in 3D simulations is merely a perturbation to the original base state. As the figure shows, there is good agreement for both sets of contours. Figure 5.3 contains plots of the azimuthal velocity and the electric potential to show that the $m = 2$ stationary mode is indeed found for $Ha = 10$, as was predicted by linear stability theory. The same contour comparisons for $Ha = 5$ are also made for $Ha = 10$. Once again, the contours for the azimuthal velocity and the electric potential from the 3D numerical simulations show good agreement with the contours from the linear stability analysis.



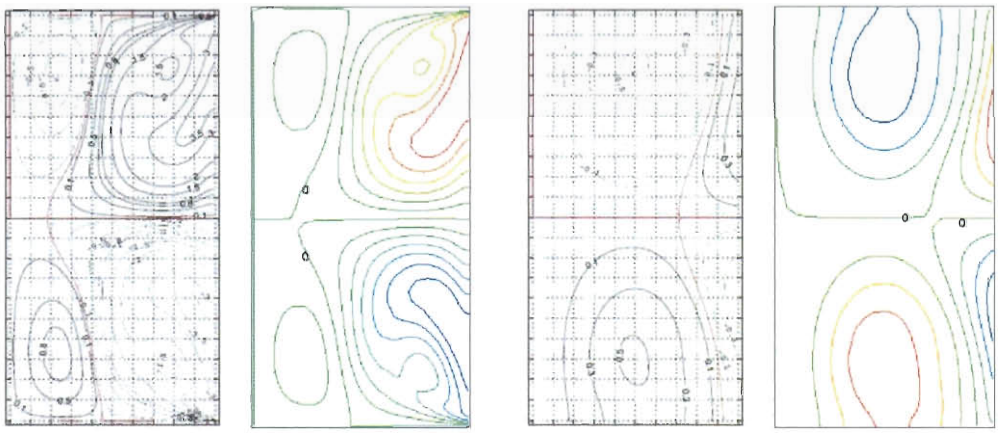
(a) (b) (c) (d)

Figure 5.9 : A comparison of 3D simulations to linear stability theory for the $Ha = 5$, $m = 2$ modes via contours of a) the azimuthal velocity from linear stability, b) the azimuthal velocity from 3D simulations, c) the electric potential from linear stability, and d) the electric potential from 3D simulations. The azimuthal velocity from 3D simulations contours are at $1.13n$ for $n = -4, \ldots, 4$ and the electric potential contours are at 0.113 for $n = -7, \ldots, 6$.



(a) (b)

Figure 5.10 : The stationary $m = 2$ mode for $Ha = 10$ at $Re_{tc} = 2750$ shown in plots of a) the azimuthal velocity and b) electric potential.



(a) (b) (c) (d)

Figure 5.11 : A comparison of 3D simulations to linear stability theory for the $Ha = 10$, $m = 2$ modes via contours of a) the azimuthal velocity from linear stability, b) the azimuthal velocity from 3D simulations, c) the electric potential from linear stability, and d) the electric potential from 3D simulations. The contours from the 3D simulations for the azimuthal velocity are at 0.972 for $n = -6, \ldots, 6$ and the contours for electric potential are at 0.244 for $n = -3, \ldots, 3$.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

We have developed a fluid flow and heat transfer solver based on the spectral element method. The methods and implementation details have been discussed and the solver has been benchmarked using four different problems. The solver is general purpose with a hard-coded axial magnetic field for liquid bridge simulations.

For the thermocapillary liquid bridge, the 3D numerical simulations matched the results of the linear stability analysis well in predicting the critical thermocapillary Reynolds number, overpredicting by only $\sim 3 - 6\%$. The $m = 2$ antisymmetric stationary mode was found for $Re_{tc} = 1650$, but a competition between the $m = 2$ stationary antisymmetric and symmetric modes was found for $Re_{tc} = 1700$ and $Re_{tc} = 1750$, which made the flow appear to be an $m = 2$ periodic mode instead. The azimuthal velocity for each mode matched the azimuthal velocity perturbations from the linear stability analysis very closely.

Magnetic stabilization proved to damp the flow such that the instability was delayed. All $Ha > 0$ cases were found to be in good agreement with linear stability theory in finding the mode number and the azimuthal velocity flow distribution.

6.2 Future Work

Advances can be made in the solver so that it is more efficient both in memory usage and in computational time. One way to greatly reduce the memory usage is to move to a matrix-free implementation of the solver. Most of the memory used by the code is in storing all of the matrices so freeing up the memory would allow for larger simulations.

Computational time may increase by using different solution techniques. Solving the fully coupled system via implicit time-stepping is expensive. One option is to use an explicit scheme for the nonlinear portion of the system and an implicit scheme for the linear portion, which will yield a linear system at each time step so that the Newton-Raphson iterations are not necessary. Examples of these methods can be found in [9, 20]. Also, the terms of the equations could be integrated in substeps typical of high-order splitting schemes [21]. This would improve the speed of the code since the sub-problems are smaller and consist of Poisson and Helmholtz equations, for which fast solvers and appropriate preconditioners have been developed [9, 20]. Finally, rewriting some of the more computationally intensive routines such as gmres could have a significant impact on the code's overall performance.

One way to gain efficiency in both memory usage and computational time is to fully parallelize the code. Efforts have been made to parallelize the majority of the code, but there still remains some much needed parallelization in the preconditioning before the code can perform adequately in parallel.

For the liquid bridge problem, we would like to continue the simulations up to higher Hartmann and thermocapillary Reynolds numbers and continue comparing to linear stability theory. We would also like to visualize the periodic instability in the full-zone much like we did in the half-zone. Other parameters such as the Prandtl number can also be altered and simulated.

One interesting extension of the problem is to apply a rotating magnetic field. Linear stability theory suggests that should delay the instability and stir the flow so that the resulting crystal has more uniform properties.

Bibliography

[1] B. Houchens and J. Walker, "Modeling the Floating Zone: Instabilities in the Half Zone and Full Zone," *Journal of Thermophysics and Heat Transfer*, vol. 19, no. 2, pp. 186–198, 2005.

[2] B. Houchens, *Multi-Scale Model of the Stability of a Laterally Heated Liquid Bridge Under the Influence of an Axial Magnetic Field.* PhD thesis, University of Illinois at Urbana-Champaign, 2005.

[3] Y. Huang, "High resolution numerical study of a liquid bridge marangoni flow with applied axial magnetic field for low prandtl number fluids," Master's thesis, Rice University, 2009.

[4] M. Levenstam and G. Amberg, "Hydrodynamical instabilities of thermocapillary flow in a half-zone," *Journal of Fluid Mechanics*, vol. 297, pp. 357–372, 1995.

[5] A. Patera, "A spectral element method for fluid dynamics: laminar flow in a channel expansion," *Journal of Computational Physics*, vol. 54, no. 3, pp. 468–488, 1984.

[6] I. Babuška, "The finite element method with Lagrangian multipliers," *Numerische Mathematik*, vol. 20, no. 3, pp. 179–192, 1973.

[7] F. Brezzi, "On the existence, uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers," *RAIRO Anal. Numer*, vol. 8, no. 2, pp. 129–151, 1974.

[8] Y. Maday and A. Patera, "Spectral element methods for the incompressible Navier-Stokes equations," in *IN: State-of-the-art surveys on computational mechanics (A90-47176 21-64). New York, American Society of Mechanical Engineers, 1989, p. 71-143. Research supported by DARPA.*, vol. 1, pp. 71–143, 1989.

[9] G. Karniadakis and S. Sherwin, *Spectral/hp element methods for computational fluid dynamics*. Oxford University Press, USA, 2005.

[10] Y. Huang and B. Houchens, "Magnetic stabilization, transition and energy analysis in the marangoni driven full-zone at low prandtl numbers," *European Physics Journal Special Topics*, vol. 192, pp. 47–62, 2011.

[11] Q. Chen and I. Babuska, "Approximate optimal points for polynomial interpolation of real functions in an interval and in a triangle," *Computer Methods in Applied Mechanics and Engineering*, vol. 128, no. 3-4, pp. 405–417, 1995.

[12] C. Catlett, W. Allcock, P. Andrews, R. Aydt, R. Bair, N. Balac, B. Banister, T. Barker, M. Bartelt, P. Beckman, *et al.*, "Teragrid: Analysis of organization, system architecture, and middleware enabling new types of applications," *HPC and Grids in Action, Amsterdam*, 2007.

[13] W. Gordon and C. Hall, "Construction of curvilinear co-ordinate systems and applications to mesh generation," *International Journal for Numerical Methods in Engineering*, vol. 7, no. 4, pp. 461–477, 1973.

[14] W. Gordon and C. Hall, "Transfinite element methods: blending-function interpolation over arbitrary curved element domains," *Numerische Mathematik*, vol. 21, no. 2, pp. 109–129, 1973.

[15] A. Squillacote, *The ParaView guide: A Parallel visualization application.* Kitware, 2006.

[16] H. Ku, R. Hirsh, and T. Taylor, "A pseudospectral method for solution of the three-dimensional incompressible Navier-Stokes equations," *Journal of Computational Physics*, vol. 70, no. 2, pp. 439–462, 1987.

[17] G. Guj and F. Stella, "A vorticity-velocity method for the numerical solution of 3D incompressible flows," *Journal of Computational Physics*, vol. 106, no. 2, pp. 286–298, 1993.

[18] L. Kovasznay, "Laminar Flow Behind a Two-Dimensional Grid," in *Proceedings of the Cambridge Philosophical Society: Mathematical and physical sciences*, pp. 58–62, Cambridge Philosophical Society, 1948.

[19] H. Blackburn and S. Sherwin, "Formulation of a Galerkin spectral element-Fourier method for three-dimensional incompressible flows in cylindrical geometries," *Journal of Computational Physics*, vol. 197, no. 2, pp. 759–778, 2004.

[20] W. Couzy, *Spectral element discretization of the unsteady Navier-Stokes equations and its iterative solution on parallel computers.* PhD thesis, École Polytechnique Fédérale de Lausanne, 1995.

[21] G. Karniadakis, M. Israeli, and S. Orszag, "High-order splitting methods for the incompressible Navier-Stokes equations," *Journal of Computational Physics*, vol. 97, no. 2, pp. 414–443, 1991.

Appendix A

Algorithms

The algorithms given here correspond to solving a linear system

$$Ax = b \tag{A.1}$$

with left and right preconditioners M_l and M_r, respectively.

A.1 Preconditioned GMRES

$r_0 = M_l^{-1}(b - Ax_0)$, $\beta = ||r_0||_2$, and $v_1 = r_0/\beta$

For $i = 1, \ldots, m$

$w = M_l^{-1} A M_r^{-1} v_i$

For $k = 1, \ldots, i$

$h_{k,i} = w^T v_k$

$w = w - h_{k,i} v_i$

$h_{i+1,i} = ||w||_2$

$v_{i+1} = w/h_{i+1,i}$

$V_m = [v_1, \ldots, v_m]$

$\hat{H}_m = \{h_{i,j}\}_{1 \leq i \leq j+1;\ 1 \leq j \leq m}$

$y_m = \text{argmin}_y \left|\left|\beta e_1 - \hat{H}_m y\right|\right|_2$

$x_m = x_0 + M_r^{-1} V_m y_m$

A.2 Preconditioned BiCGStab

$r_0 = b - Ax_0$

Choose $\tilde{r}$ such that $\tilde{r}^T r_0 \neq 0$

For $i = 1, \ldots, i_{max}$

$\rho_i = \tilde{r}^T r_{i-1}$

If $i = 1$

$p_i = r_{i-1}$

Else

$\beta = (\rho_i / \rho_{i-1})(\alpha \omega_{i-1})$

$p_i = r_{i-1} + \beta(p_{i-1} - \omega_{i-1} v_{i-1})$

$\hat{p} = M^{-1} p_i$

$v_i = A\hat{p}$

$\alpha = \rho_i / (\tilde{r}^T v_i)$

$s = r_{i-1} - \alpha v_i$

$\hat{s} = M^{-1} s$

$t = A\hat{s}$

$\omega_i = t^T s / t^T t$

$x_i = x_{i-1} + \alpha_i \hat{p} + \omega_i \hat{s}$

$r_i = s - \omega_i t$

If $||r_i|| < tol$, then quit

A.3 ILU(0) Preconditioner

For $i = 2, \ldots, n$

  For $k = 1, \ldots, i-1$

    If $(i,k) \in NZ(A)$

      $a_{ik} = a_{ik}/a_{kk}$

      For $j = k+1, \ldots, n$

        If $(i,j) \in NZ(A)$

          $a_{ij} = a_{ij} - a_{ik}a_{kj}$